

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

INFORMATION ENGINEERING OF THE CURRICULAR
OFFICERS' SEGMENT OF A UNIFIED STUDENT ACADEMIC
DATABASE SYSTEM FOR NPS

by

Michael S. Haas
and
Mary L. Hochstetler

September, 1991

Thesis Advisor:

Professor Daniel R. Dolk

Approved for public release; distribution is unlimited

T260863

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School			
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
8c ADDRESS (City, State, and ZIP Code)		10 SOURCE OF FUNDING NUMBERS			
		Program Element No	Project No	Task No	Work Unit Accession Number
11. TITLE (Include Security Classification) Information Engineering of the Curricular Officers' Segment of a Unified Student Academic Database for NPS Unclassified					
12. PERSONAL AUTHOR(S) Captain Michael S. Haas, United States Marine Corps and Captain Mary L. Hochstetler, United States Marine Corps					
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED From To	14. DATE OF REPORT (year, month, day) 1991, September, 26	15. PAGE COUNT 342		
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17. COSATI CODES			18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUBGROUP	Information Engineering (IE), Integrated Computer-Aided Software Engineering (I-CASE), Information Engineering Facility (IEF), Information Strategy Planning (ISP), Business Area Analysis (BAA)		
19. ABSTRACT (continue on reverse if necessary and identify by block number) The Naval Postgraduate School (NPS) plans to develop the Unified Student Academic Database (USAD), using an Integrated Computer-Aided Software Engineering (I-CASE) tool. USAD is intended to consolidate requirements of the Director of Programs, Registrar, Admissions Office, and Curricular Officers. The current strategy for utilizing an I-CASE tool at NPS is sub-optimal. Texas Instrument's (TI's) Information Engineering Facility (IEF), was purchased to conduct analysis and design of USAD. IEF is designed to commence with a thorough analysis of an organization's Information Strategy Plan (ISP). However, TI proclaims ISP is not essential. An investigation was conducted into the advisability of omitting the ISP phase at NPS. The Curricular Officers' USAD requirements were modeled commencing with the Business Area Analysis (BAA), the second stage of IEF. This thesis determined bypassing the ISP phase for USAD would be inappropriate. Furthermore, using I-CASE tools for a project's front-end management only is not recommended. Payback is realized only when an organization commits to a full-scale strategic I-CASE implementation plan.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS REPORT <input type="checkbox"/> LIMIT USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Professor Daniel R. Dolk			22b. TELEPHONE (Include Area code) (408) 646-2260		22c. OFFICE SYMBOL AS/Dk

Approved for public release; distribution is unlimited.

Information Engineering of the Curricular Officers' Segment
of a Unified Student Academic Database for NPS

by

Michael S. Haas
Captain, United States Marine Corps
B.S., Cornell University

and

Mary L. Hochstetler
Captain, United States Marine Corps
B.A., Messiah College

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

ABSTRACT

The Naval Postgraduate School (NPS) plans to develop the Unified Student Academic Database (USAD), using an Integrated Computer-Aided Software Engineering (I_CASE) tool. USAD is intended to consolidate requirements of the Director of Programs, Registrar, Admissions Office, and Curricular Officers. The current strategy for utilizing an I_CASE tool at NPS is sub-optimal. Texas Instrument's (TI's) Information Engineering Facility (IEF), was purchased to conduct analysis and design of USAD. IEF is designed to commence with a thorough analysis of an organization's Information Strategy Plan (ISP). However, TI proclaims ISP is not essential. An investigation was conducted into the advisability of omitting the ISP phase at NPS. The Curricular Officers' USAD requirements were modeled commencing with the Business Area Analysis (BAA), the second stage of IEF. This thesis determined bypassing the ISP phase for USAD would be inappropriate. Furthermore, using I_CASE tools for a project's front-end management only is not recommended. Payback is realized only when an organization commits to a full-scale strategic I_CASE implementation plan.

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	PROBLEM DESCRIPTION	1
B.	RESEARCH QUESTIONS	2
C.	INVESTIGATIVE METHODOLOGY	3
D.	SCOPE AND ASSUMPTIONS	4
E.	STRUCTURE OF THESIS	6
II.	BACKGROUND	7
A.	HISTORY OF CURRENT STUDENT DATABASE SYSTEMS . .	7
B.	CURRICULAR OFFICERS' ANXIETY	10
III.	LITERATURE REVIEW	12
A.	PREVIOUS ANALYSES	12
B.	BRIEF OVERVIEW OF SYSTEM DEVELOPMENT METHODOLOGIES	14
1.	Information Engineering	14
2.	Comparison of IE and Yourdon Structured Methodology	16
3.	Computer Aided Software Engineering (CASE) Tools	18
4.	IEF's Position in the I_CASE Market	21

IV.	INVESTIGATION STRATEGY	22
A.	OBTAINING IEF EXPERIENCE	22
	1. Vendor Documentation	22
	2. Vendor Training and Technical Support . . .	22
	3. Hands-On	23
B.	INTERVIEWS OF IEF USERS	24
	1. Description of Process	24
	2. Questionnaire	25
V.	PRESENTATION OF DATA COLLECTED	26
A.	TRENDS DETERMINED DURING USER INTERVIEWS . . .	26
	1. ISP	26
	2. Productivity Increases Cited	31
	3. Type of Projects	32
	4. Plaudits	32
	5. Warnings/Complaints	34
B.	IEF DEFICIENCIES NOTED DURING DEVELOPMENT . . .	35
	1. Report Generation	35
	2. Graphical User Interface (GUI)	35
	3. Limited Languages Developed	36
	4. Training/Learning Curve	37
VI.	ANALYSIS OF CURRICULAR OFFICER REQUIREMENTS . . .	38
A.	MODIFICATION TO EXISTING REQUIREMENTS ANALYSIS	38
B.	DEVELOPMENT OF CURRICULAR OFFICER REQUIREMENTS	
	MODEL	40

VII. CONCLUSIONS/RECOMMENDATIONS	44
A. BASED ON INTERVIEW	44
B. BASED ON EXPERIENCE	46
C. BASED ON TI'S IEF PERSONNEL	47
D. TRAINING (IE & IEF)	48
1. Based on Interviews	48
2. Based on Experience	50
E. IE METHODOLOGY AS COURSE AT NPS	50
F. RESTRICTED ACCESS	51
G. ACCOUNTABILITY/OWNERSHIP OF DATA ELEMENTS	52
H. LIMITING IEF'S POTENTIAL	52
APPENDIX A	54
APPENDIX B	58
APPENDIX C	86
APPENDIX D	90
APPENDIX E	108
APPENDIX F	139
APPENDIX G	230

APPENDIX H	233
APPENDIX I	300
APPENDIX J	315
LIST OF REFERENCES	330
INITIAL DISTRIBUTION LIST	333

I. INTRODUCTION

A. PROBLEM DESCRIPTION

The Naval Postgraduate School (NPS) is responsible for graduate level education of military officers from both domestic and foreign military services. Roughly 2000 students receive training annually at NPS. Tracking these students from their point of entry at the school, through their respective programs until graduation, generates a substantial amount of information.

NPS currently maintains three functionally separate database systems which contain an enormous amount of duplicated information pertaining to student records. Reconciliation of these database files must be conducted frequently to ensure data integrity is maintained.

In order to eliminate this duplication of effort and inconsistency of files, the development of a consolidated system, the Unified Student Academic Database (USAD)¹, was proposed by the Associate Dean of Faculty and Graduate Studies in February 1990 [Ref. 1]. Other perceived benefits from this integration include the ability to obtain more information from the data and to perform thorough data and

¹ The acronym "USAD" has not been officially designated by NPS as the approved title for this project.

trend analysis. A current problem which would be rectified by such a consolidated system is the provision of a single reliable source for the real-time accountability of enrolled students. The elimination of redundant effort and assurance of data accuracy and integrity are of major concern to the efficient conduct of business at the Naval Postgraduate School. [Ref. 2]

B. RESEARCH QUESTIONS

The NPS' Management Information Systems (MIS) Department was appointed to conduct a detailed requirements analysis of an integrated system. The MIS Department purchased Texas Instrument's (TI) integrated computer-aided software engineering (I_CASE) tool, Information Engineering Facility (IEF) to aid in this process. [Ref. 3]

IEF is based on the seven basic building blocks of Information Engineering (IE) as developed by James Martin and Clive Finkelstein. [Ref. 4] The first phase of IE produces a high-level organizational blueprint called Information Strategic Planning (ISP). Subsequent design of specific business functions commences in the Business Area Analysis (BAA) phase of IE. Ideally ISP is conducted prior to BAA modeling to provide "project continuity and insure that the scope of the BAA project is correctly set" within the corporate information strategy [Ref. 5:p. 117].

TI's documentation indicated that organizations with immediate needs in a particular business area could bypass the ISP and proceed directly with the BAA [Ref. 6:p. 14]. The Director of MIS envisioned such an approach to rapidly model and document the intended USAD.

Therefore, the primary focus of this thesis is to determine the viability of analyzing and designing the Curricular Officers' requirements for the USAD system at NPS commencing with the BAA instead of the ISP. Such an analysis is expected to also yield answers to the following questions:

1. What are the USAD specifications, based upon Curricular Officers' requirements?
2. How are Curricular Officers to be restricted in accessing/utilizing data elements not considered under their cognizance?
3. How is accountability/ownership of data elements to be established?

C. INVESTIGATIVE METHODOLOGY

Three approaches were used to investigate the above research questions. First hand accounts of projects developed using IEF were gathered from actual users of the tool, external to NPS. A semi-structured interview process was utilized to enable them to elaborate freely on their professional experiences. A second approach involved the research of publications and vendor documentation concerning the use of the IEF tool. Finally, direct personal experience

was gained through laboratory experimentation and vendor-supplied training with the IEF tool.

D. SCOPE AND ASSUMPTIONS

The original scope of this thesis was to produce specifications of the entire USAD integrated system, designed to maintain and manipulate student academic records for the Director of Programs, Admissions, Registrar and Curricular Officers, using IEF. However, due to MIS' long term development plans and the requirement to avoid possible interference with current development efforts in the Registrar's office, the scope of this research was narrowed to address only requirements of the Curricular Officers.

The narrowed scope was a necessary compromise to ensure cooperation from controlling departments. Initially, it seemed as if this thesis might be stillborn because the Director of MIS was concerned that students might stir up unrealistic user expectations. It seems that the Director had previously found it necessary to exert an inordinate amount of time and energy lowering the users' expectations for near-term delivery of USAD following two students' 1990 requirements study [Ref. 7]. Although this study was only intended as an initial analysis, and not part of the overall MIS strategy for producing USAD, users were convinced that the new system was in production. After two summit meetings and an agreement to focus on the use of IEF vice the production of

specifications for USAD, the Director of MIS agreed to allow a student team to commence work on modelling the Curricular Officers' requirements [Ref. 8].

The automated design model produced in this thesis reflects current Curricular Officer requirements, as detailed in Reference 7, and has been developed in the existing tool planned for implementation by the MIS department at NPS. This design should decrease the time and effort required by the MIS staff to fully design and implement a system, yielding both fiscal savings and increased productivity.

Assumptions were made during the course of development of the Curricular Officer's portion of USAD to enable the IEF tool to be employed effectively. Without the advantage of an ISP, some business processes modeled were therefore created from necessity and were not verified by users. They may not be in accordance with existing policies. These processes, which affect entity attributes not under the purview of the Curricular Officers, must exist to enable the Curricular Officer's BAA model to be completed within the strict methodology of the tool. None of these constructions materially effected the examination of the central research question. However, verification and validation of these processes will be required prior to designing the entire USAD.

E. STRUCTURE OF THESIS

Chapter II explains the background of the present system utilized by NPS and independent programming efforts undertaken by the Curricular Officers to counteract their frustration with that system.

Chapter III provides an overview of previous efforts to identify Curricular Officer USAD requirements. A discussion of Information Engineering (IE) and its comparison to Yourdon's Structured Methodology (YSM) is also provided. Finally, IEF's relative standing in the I_CASE environment is addressed.

Chapter IV highlights the strategies employed during the course of this research to investigate the utility of IEF.

Chapter V documents the accounts obtained from interviews with current IEF users in the private, federal, and DoD communities. Hands-on development experience with IEF is also discussed.

Chapter VI presents necessary modifications to the previous analysis of the Curricular Officers' requirements. Explanation is provided for IEF output reports and diagrams of the data and activity models for USAD.

Chapter VII summarizes the conclusions and recommendations.

II. BACKGROUND

A. HISTORY OF CURRENT STUDENT DATABASE SYSTEMS

An account of past actions is necessary to provide a clear picture of the present Curricular Officers' requirements. In 1985 a FOCUS™ application was implemented on the NPS mainframe computer in an attempt to maintain the enormous amount of student academic information. Figure 1 illustrates this flow of information between the primary offices involved.

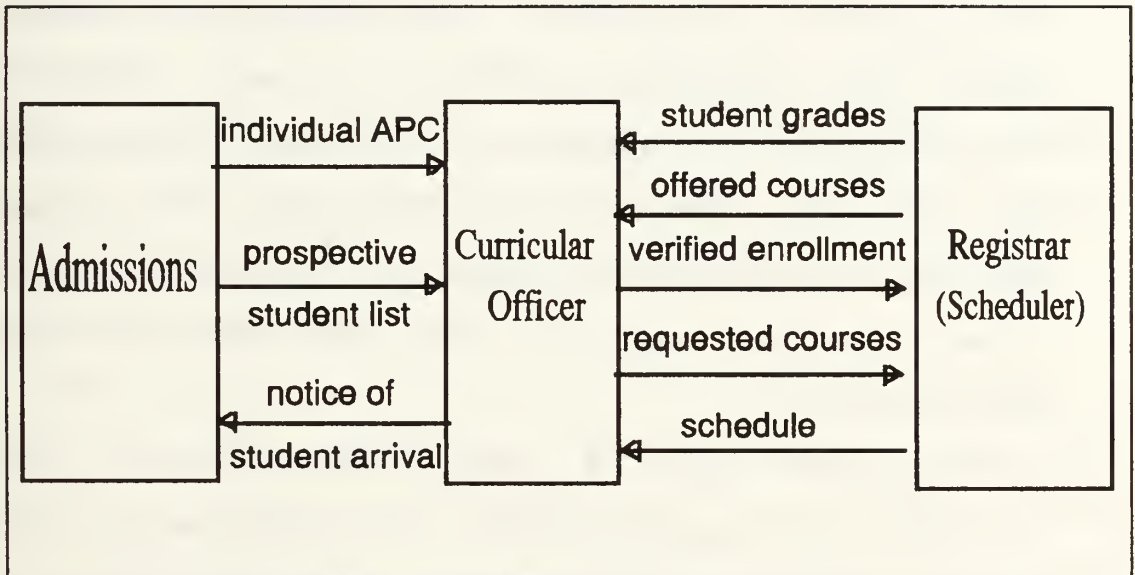


Figure 1. Information Flow

As students progressed through the postgraduate process, the Registrar's office generated student information. This information was made available for use by the Curricular Officers to monitor their students' enrollment and academic status. [Ref. 9:p. 3]

Eventually, automated access to the Registrar's files was removed for reasons we could not find documented. Special requests had to be made for reports to allow Curricular Officers to update their records. Such restrictions instigated the creation of a separate Curricular Officer database. [Ref. 9:pp. 3, 7]

The Admissions database was created shortly thereafter. Admissions would input prospective student data upon receipt of military orders from higher headquarters. Two reports were written for the Curricular Officers' use. The first would show prospective students for whom orders were received; the second would verify the arrival and enrollment of new students which could then be loaded into a Curricular Officer database. The Curricular Officer was responsible for reconciling redundant differences among separate databases (printouts of student records would be provided for verification/modification).

The current system is virtually unmaintainable. Inability of current in-house programmers to modify the present FOCUS system influenced the recommendation to develop a new system. Potential ripple effects of additional manipulation to source code could have disastrous effects transparent to well-intentioned programmers. The MIS staff has recently received requisite FOCUS programmer training in an effort to resolve problems associated with maintaining the existing FOCUS system, until USAD can be developed using IEF. [Ref. 2]

Lack of user support and training over the years has prevented the development of the FOCUS system's true potential. The structure in place was built from specifications which were not intended to fully coordinate all users' concerns into a consolidated system. Thorough analysis was short-circuited in an attempt to meet users' immediate needs. [Ref. 2]

Users indicated a feeling of little control over files they were processing. Therefore, a tendency to avoid and/or bastardize the use of the system began to manifest itself, especially in Curricular Offices. Frustration with the complexity and lack of user-friendliness forced users to develop their own systems. "Some are using the same systems set up in 1985, last modified in 1986, and others have updated their systems based on their knowledge and interest in computer programming." [Ref. 9:p. 7]

Additionally, since the Curricular Officers were denied access to the Admissions' and Registrar's databases "Curricular Offices were writing their own programs in FOCUS and, AT THE SAME TIME, creating additional databases on PC's to complement the FOCUS system. This is a major duplication of effort." [Ref. 9:p. 7]

High turnover rate of office personnel further hampered the present system, since their departure depleted the level of corporate knowledge derived from original training offered with this system. Additionally, documentation needed to

navigate personnel through the system was inadequate or nonexistent. [Ref. 10]

B. CURRICULAR OFFICERS' ANXIETY

Multiple Curricular Officer requests were made to the MIS Department for modifications and assistance in the use of the Curricular Officer's information system. In December 1990, a meeting was conducted with the MIS Department and Curricular Officers to address their immediate needs. It was agreed that at the very least, the Curricular Officers required read-only access to the Registrar's files to enable Curricular Officers to identify, supervise and counsel students whose academic standing required attention. [Ref. 11]

Eventually, some Curricular Officers pressed for a more responsive and effective student information system, which prompted Lieutenant Aaron Rouska and Lieutenant Commander Eric VanNortwick to conduct a requirements study in May 1990. [Ref. 7] The overriding concern that prompted the initiation of this study was the Curricular Officers' desires to ensure that their specific requirements would be included in the forthcoming USAD. This well-documented research provided foundation for the analysis portion of this thesis in beginning the IEF modeling process at the BAA level.

The Dean of Computers and Information and the Director of MIS indicated that the 1990 documentation would be sufficient for this purpose. While this was generally found to be the

case, analysis "gaps" existed, such as the specific role of the Curricular Officer in creating records for students who were not projected students and therefore not created by the Admissions Office. This type of missing detail hindered completion of the modeling effort within the IEF Design toolset, requiring additional interviews with a representative for the Curricular Officers to clarify specific aspects of data handling, functional processes and entity relationships. [Ref. 12]

The background of the current system lays part of the framework in which this analysis is conducted. An overview of prior studies and the workings of IEF are required to provide a common understanding of both user requirements and the theoretical structure of the I_CASE tool.

III. LITERATURE REVIEW

A. PREVIOUS ANALYSES

An attempt was made by the MIS department in January 1988, to identify the purpose, responsibilities and requirements of the Admissions, Registrar, Scheduler, and Curricular Offices for a student records database. Results of that analysis were documented in the Student and Academic Records System (STARS) report composed by Mr Michael Spencer. This report "served as the foundation for identifying and validating the information needs of the Curricular Officers" in the Rouska and VanNortwick study of May, 1990. [Ref. 7:p. 1]

The studies indicated that the role of the Curricular Officer involved four areas: sponsor liaison, curricular development and management, student supervision and counseling, and resource management. The studies' analyses specified automated support for only the student supervision and counseling portions of the Curricular Officers' responsibilities.

The following processes were outlined in the Rouska and VanNortwick study [Ref. 7] as required elements to model activities in which the Curricular Officer is involved directly or peripherally:

1. Evaluating Prospective Students
 - a. Calculating APC (Admissions)
 - b. Evaluating acceptance
 - c. Processing prospective students (Admissions)
2. Maintenance of Academic Records
 - a. Updating records (Registrar)
 - b. Approving thesis (Thesis Processor)
 - c. Administering grades (Registrar & Academic Offices)
3. Scheduling Courses
 - a. Course/Professor scheduling (Academic Departments)
 - b. Course registration
 - c. Creation of Exceptions (Registrar)
4. Supervision of Students
 - a. Check-in new students
 - b. Student supervision
5. Generating Reports

The above processes indicate areas of overlap between various entities at NPS. The specific requirements of the Curricular Officer are more evident in the following procedures performed by that office:

1. Welcome Aboard/Student Check-in
 - a. accepts the student
 - b. ensures student sponsor is identified & student notified
 - c. sends welcome aboard package
 - d. ensures student completes check-in
2. Academic Scheduling
 - a. validation process for qualified students
 - b. sets up initial course of study (templates)
3. Academic Forecasting/Programming
 - a. maintains/changes individual student course of study (templates)
 - b. request course through program cards
4. Curricular Officer files, records and reports
 - a. submission student fitness reports
 - b. notification of degrees

5. Thesis completed or extensions requested
6. Reports of Academic Performance/Progress
 - a. stores, monitors individual student progress
 - b. compiles list of graduating students

These procedures were used in the creation of the IEF data and activity model outlined in the appendices of this thesis.

B. BRIEF OVERVIEW OF SYSTEM DEVELOPMENT METHODOLOGIES AND AUTOMATED TOOLS

1. Information Engineering

Information Engineering is the underlying methodology of IEF. IE provides a comprehensive framework for satisfying information needs of an organization by dividing the system development process into stages. There are seven stages of IE, as shown in Figure 2:

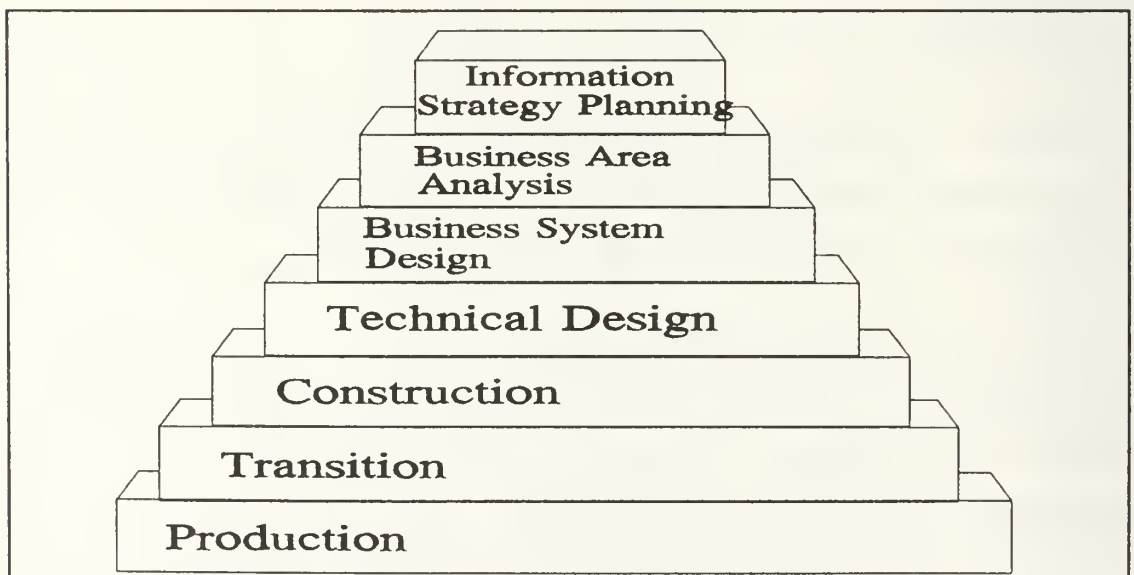


Figure 2. Seven Stages of Information Engineering
[Ref. 6:pp. 3-4]

1. **Information Strategic Planning (ISP)** provides an opportunity for organizational planners to elucidate a broad framework of information requirements of the entire business. Such a plan requires top level management involvement. During this process an overarching blueprint is produced from which smaller subdivisions can be derived.
2. **Business Area Analysis (BAA)** is the stage in which a specific segment of the organization (called a **business area**) is evaluated by analysts to develop a more restricted conceptual model of what occurs in this one business area, based on its peculiar information requirements.
3. **Business System Design (BSD)** involves fashioning details of how the user will interface with the developed system application (i.e. - business system). Designers are concerned solely with the man-machine interface and ignore the intended computing platform in this phase.
4. **Technical Design (TD)** is the first phase where designers become concerned about the targeted computing environment. The hardware, operating system, and database management system (DBMS) are all considered in tailoring results of the BSD to fit this environment.
5. **Construction** is the stage in which developers produce a fully executable application that can be run in the targeted computing environment. Components generated include processes, job control statements, screen formats, and transaction definitions.
6. **Transition** is the installation of the constructed system in its production environment. Installation may involve replacement of all or part of the existing system.
7. **Production** is when the business begins to experience wide range of benefits derived from capabilities of the application system under execution. Needs and requirements modeled during ISP and BAA are being satisfied with the existing application system.

2. Comparison of IE and Yourdon Structured Methodology

Many of the current generation of Information Technology (IT) professionals have been trained in conventional techniques of the Yourdon Structured Methodology (YSM). They are quite comfortable drawing data flow diagrams (DFDs) and structure charts based on this training. [Ref. 13:p. 1] IEF does not support YSM, but rather James Martin's and Clive Finkelstein's Information Engineering (IE) Methodology.

Texas Instrument's Tamer Uluakar compares the two methodologies and succinctly highlights cogent differences between the two. [Ref. 13] Figure 3 provides a comparison summary between the two methodologies - IE, as practiced in TI's IEF, and YSM. The following brief explanation germane to the research question at hand is presented without embellishment.

IE and YSM lifecycles are generally similar with several notable differences. IE life-cycle starts with Information Strategic Planning (ISP) at the enterprise level followed by analysis of the business area of interest before focusing on a system. Business areas are defined during ISP as pieces of the enterprise which can be analyzed independent of one another. The scope of a business area should be analyzed all at once...to avoid scope creep and future system integration problems. YSM is currently lacking a strategic planning phase. In absence of the business area concept, the YSM life-cycle starts with requirements definition for a particular **system**.

In addition to this difference in scope, YSM's analysis differs from IE's 'business area analysis' in one other way. In YSM, analysis includes modelling the required processes and the flow of data in response to each event. In IE, the processes required for each event are defined

during the analysis but the dynamics of the response (ie., the flow of data among the processes if more than one process is involved) is not modelled until design. [Ref. 13:p. 6]

<p style="text-align: center;">INFORMATION ENGINEERING BY TEXAS INSTRUMENTS</p>	<p style="text-align: center;">STRUCTURED METHODOLOGY BY YOURDON</p>
<p><u>I. INFORMATION STRATEGY PLANNING</u></p> <ol style="list-style-type: none"> 1. Broad Brush View of the Enterprise <ol style="list-style-type: none"> a) Information Architecture b) Business System Architecture c) Technical Architecture <p><u>II. BUSINESS AREA ANALYSIS</u></p> <ol style="list-style-type: none"> 2. Detailed Essential Requirements for a Business Area <ol style="list-style-type: none"> a) Data Model b) Functional Decomposition (Object Life-Cycle Partitioned) c) Dependency Analysis d) For Elementary Processes: <ul style="list-style-type: none"> - Inputs - Outputs - External Objects - Events - Process Specifications (PAD) 3. Identification of Business Systems for the Business Area <p><u>III. BUSINESS SYSTEM DESIGN</u></p> <ol style="list-style-type: none"> 4. Event Response Modeling (not formalized) 5. Packaging of the Process Groupings into Procedures to Support a Shared User Interface (e.g., a Screen) 6. Procedure and Dialog Flow Design 7. Interface Design <p><u>IV. TECHNICAL DESIGN</u></p> <ol style="list-style-type: none"> 8. Database Design (mostly automatic) 9 - 10. (Not needed) 11. Load Module Packaging <p><u>V. CONSTRUCTION</u></p> <ol style="list-style-type: none"> 12. Database Generation (automatic) 13. Programming, Compilation, and Installation (automatic) 14. Testing (against procedure specs) 	<p><u>I. STRATEGY PLANNING</u></p> <ol style="list-style-type: none"> 1. (Not formally included in the methodology) <p><u>II. ANALYSIS</u></p> <ol style="list-style-type: none"> 2. Detailed Essential Requirements for a System <ol style="list-style-type: none"> a) Environmental Model <ul style="list-style-type: none"> - Context Diagram - Events List b) Data Model c) Functional Decomposition (Event partitioned) d) For All Processes (using DFDs): <ul style="list-style-type: none"> - Inputs - Outputs e) For Event-Level Processes (using DFDs): <ul style="list-style-type: none"> - External Objects f) For Primitive Processes: <ul style="list-style-type: none"> - Process Specifications 3. (Not applicable) 4. Event Response Modeling (not formalized) <p><u>III. BUSINESS SYSTEM DESIGN</u></p> <ol style="list-style-type: none"> 5. The Processor Model 6. Procedure and Dialog Flow Design 7. Interface Design 8. Database Design 9. Transformation into Structure Charts 10. Module Specification 11. Load Module Packaging <p><u>IV. CONSTRUCTION</u></p> <ol style="list-style-type: none"> 12. Database Generation 13. Programming, Compilation, and Installation 14. Testing (against code specs)

Figure 3. Comparison of Life-Cycles Between Information Engineering and Yourdon Structured Methodology [Ref. 13:p. 5]

Although IE and YSM are based on the same principles, differences arise during the analysis phase. Experienced YSM analysts should not have difficulty adapting to the IE methodology. This thesis focuses on the role of the initial planning stage (ISP) of IE which is excluded from the YSM methodology.

3. Computer Aided Software Engineering (CASE) Tools

"CASE tools" is a phrase that has been the subject of debate, confusion and disappointment during the last few years. Defining the term can be difficult since contradictory claims are often made as to the exact boundaries associated with the term CASE. "It has become commonplace to refer to any software tool that aids system professionals to do their jobs as a 'CASE tool'." [Ref. 14:p. 10] However, CASE tools generally include five components: diagramming tools, an information repository, interface generators, code generators, and management tools. [Ref. 15:p. 260]

Unfortunately, early CASE tools were marketed promising greater potential than what they delivered. The resultant user frustration lowered CASE market growth from a high of 67% from 1988 to 1989, to a mere 20% the following year. "Chief among their frustration is a lack of integration between vendor's own front-end and back-end tools..." [Ref. 16:p. 61]

The five components mentioned above may appear in a given vendor's tool as individual, discrete elements, or two or more may be integrated into a single tool. The latter concept is a subset of CASE tools that has been coined "Integrated CASE" or "I_CASE" to reflect the increased capability of moving from one end of the toolset (the so-called "Front End") to the other (the "Back End") without

exiting a particular vendor's tool. [Ref. 15:p. 263] This is a vital step in the evolution of CASE tools, since IT groups have realized that "nonintegrated, 'point' products lead to inefficiency and a lack of automated control." [Ref. 16:p. 61] I_CASE tools have momentous potential to "redefine the paradigms of application software delivery." [Ref. 14:p. 10]

The overriding mission of IT groups is to deliver quality software to meet ever increasing user demand. This requires production of "higher quality applications faster, with more emphasis on reliability and maintainability, and less on technical elegance and efficiency." [Ref. 14:p. 13] Many IT professionals feel that the long-term solution to this productivity issue lies in the use of I_CASE tools. A Gartner Group report states that: "I_CASE offers the highest observable productivity improvements, ranging from 10 percent to 40 percent over a five-year period." [Ref. 17] Factors cited in Reference 17 as contributing to these productivity enhancements include:

1. the repository environment and comprehensive data model which support all objects and relationships.
2. common user interfaces among a tool's components which reduce the learning curve (as opposed to mixing and matching different vendors' tools).
3. the data obtained and utilized by different parts of an I_CASE tool which is handled more efficiently with few gaps or overlaps outside the requirements to meet the goals of the project at hand.
4. transformations between deliverables which are automated, precise and more reliable than loosely coupled tools.

While the reported payback period for I_CASE of three years or more is seen as a potential drawback "I_CASE is recommended for long-term, maximized productivity gains, with a correspondingly high up-front investment." [Ref. 17] The reasons cited for such lengthy break-even time frames are that the:

1. training in the methodology is critical; utilization of an I_CASE tool requires strict adherence to a particular tool's methodology.
2. time line from novice to master of the tool is estimated between three to 18 months, with the majority of users supporting a 12 to 18 month estimate.
3. majority of projects utilizing I_CASE solutions require implementation in their entirety before realizing the benefits expected from I_CASE; this results in a relatively higher initial investment outlay. [Ref. 17]

4. IEF's Position in the I_CASE Market

IEF is acknowledged to be among the leading tools in the industry and emerged as the top-ranked I_CASE tool in a user survey conducted by *Computerworld*. [Ref. 18] In a meeting with TI's IEF product specialists [Ref. 19], statistics were presented which showed IEF to have 22.1% of the worldwide market share and 39.8% of the North American market share. The latter represented a lead of ten percentage points over IEF's nearest competitor, Andersen Consulting. [Gartner Group 1989 reports were cited as the

source for these figures.] Gartner Group, Incorporated also identified IEF as "one of a handful of CASE vendors that can prosper in the 1990s." [Ref. 19]

Now that a foundation for I_CASE tools has been laid, the research conducted into the applicability of IEF at NPS is presented in the following chapters, commencing with a discussion of the strategies selected.

IV. INVESTIGATION STRATEGY

A. OBTAINING IEF EXPERIENCE

1. Vendor Documentation

IEF User's Manuals are very detailed and provide too wide a scope for novices to easily begin system development. Limited time available in which to learn how to use the product, prompted us to contact Texas Instrument's representatives for assistance. TI's support was outstanding in this regard.

2. Vendor Training and Technical Support

TI's San Francisco representative - Mr. Terry King - provided TI's recently released (February 1991), self-paced Rapid Development/Tutorial Module for beta testing which steers users through a simplified development of a software system. It did not illustrate the full extent of IEF's functions. The tutorial was intended to familiarize users with features and functions of IEF that are directly related to the design and implementation of information systems. [Ref. 20] Unfortunately, some sections of the beta test could not be performed since the tutorial was developed for beta testing in an OS/2 environment and could not take advantage of the DOS version of IEF purchased by NPS.

Additionally, TI provided approximately 32 hours of BAA segment I training. This course outlined building blocks for data and activity modeling at the BAA level. TI provides approximately 15 such training sessions of various lengths, to assist users in realizing the enormous potential of the tool. Such training is instrumental in reducing the learning curve required for effective use of this product by any user.

Lastly, TI routinely maintained personal contact to assist in the Curricular Officers' sub-section development of USAD. Periodically, site visits were conducted to aid in clarifying misunderstandings encountered with the tool. Such attention was instrumental in successfully managing the steep learning curve associated with using the IEF tool effectively.

3. Hands-On

Three calendar months of intensive effort were required for experimentation and familiarization with the IEF Analysis Toolset before consistent data and activity models could be developed. The NPS version of IEF was hosted on an IBM-compatible 386 clone. At least one Megabyte of RAM and 20 Megabytes of accessible hard disk space were required to model the Curricular Officer's BAA. Files created by the IEF system grew to sizes of greater than two Megabytes and required compression software to store backups on 1.44 MB floppy disks.

The enormous power of the tool became evident as hands-on practical application increased. However, we

realized only the proverbial "tip of the iceberg" compared to claims made in the vendor's brochures and the imposing mass of technical manuals provided for user guidance. Although substantially greater capabilities are available, one or more years of experience is required to attain expert proficiency. [Ref. 21]

B. INTERVIEWS OF IEF USERS

1. Description of Process

Interviews were conducted with current IEF users in the private sector, the Federal Government, and the Department of the Navy. In addressing the necessity for conducting an ISP, the interviews attempted to elicit experience-based opinions and ideas from current users. The anecdotal nature of this data is designed to indicate how some IEF users have been using the tool and to what effect. The relative immaturity of the I_CASE environment and the untested nature of available tools leads to a paucity of reliable information in this area. Independent documentation concerning the specific question of successfully implementing an IEF-developed system beginning at the BAA level is not available due to the relatively recent emergence of TI's IEF.

Interviews were not designed to produce a comprehensive market survey/analysis of IEF as a tool. *ComputerWorld*, in its April 22, 1991 issue, did a poll of users' satisfaction ratings for various I_CASE tools and

vendors. Texas Instruments' IEF received the highest overall rating of four industry-leading tools evaluated: CGI Systems' Pacbase, Arthur Andersen's Foundation, and Knowledgeware's IEW/ADW. IEF placed first in 12 of 19 categories. Its highest ratings were in integration of lifecycle stages, ability to increase quality, and code generation capabilities. Conversely, IEF's lowest ratings were in its ability to integrate with other vendor's tools, support for local area networks, and required training time. [Ref. 18] The proliferation of I_CASE tools in the marketplace makes this a fruitful area for future research.

2. Questionnaire

A semi-structured questionnaire was utilized to guide interviews with experienced IEF users and maintain a focus on the pertinent issues relating to the use of ISP in project development. However, interviewees were extremely willing to share their experiences at length and the information flow often strayed beyond the parameters of the questionnaire. Interviews usually evolved into a caucus among Information Technology (IT) professionals from civilian, federal, and military organizations on the impact of I_CASE, specifically IEF.

V. PRESENTATION OF DATA COLLECTED

A. TRENDS DETERMINED DURING USER INTERVIEWS

Although the number of professionals interviewed was small (7), a consensus developed on items relevant to the research question. There were no major areas of disagreement - even complaints about the tool were generally consistent.

The only divergent opinion expressed about IEF was that the ISP capability was not powerful enough! A particularly enormous undertaking being conducted by the Naval Aviation Maintenance Office (NAMO) encompasses the entire Navy's aviation community. This project is being considered for adoption by DoD as part of the Corporate Information Management (CIM) initiative and therefore, must be constructed with even wider strategic considerations. Arthur Andersen's *Foundation* toolset is consequently being purchased by NAMO for its front-end strategic planning capability. The current plan is to conduct strategic planning using *Foundation* and then import the output of that toolset into IEF to execute the rest of the project.

1. ISP

Whether or not to undertake an ISP seemed to hinge on the size, scope, and project definition. All users uniformly agreed that if the project were bounded and sufficiently

constrained, then the ISP was not necessary. The ISP stage is not required in an isolated business process where the following are all well known and clearly defined: 1) the data being accessed, 2) the processes involved with timing and coordination of data access, 3) the intricacies of business relationships, and 4) the business rules and policies affected by these data flows and processes.

When a project crosses functional areas however, the requirement for an ISP increases substantially. If a particular functional area can not be modeled in isolation, some form of ISP is required to capture top management's perspective. Additionally, the ISP provides a tool for obtaining coordination and agreement of various functional area managers prior to attempting to model the business area.

The ISP provides a management tool for critically analyzing existing goals and functions of an organization. The advantages of this process are therefore available to other than IEF users in an organization. In an in-house, organization-wide ISP conducted by NAMO,² results of the ISP affected the eventual decision that several "rice bowls" could be consolidated.

Most users reported that they did not initially utilize ISP in their pilot projects conducted with IEF. This

² Reported by Mr. Joe Joseph - the Computer Systems Analyst primarily responsible for the conduct of ISP at Naval Aviation Maintenance Office (NAMO), Naval Air Station Patuxent River, MD

hesitance was a result of careful consideration of the political climate, rather than lack of faith in the concept or importance of ISP. Most initiated relatively small pilot projects with IEF at the BAA level in order to produce a quick victory to promote use of the tool organization-wide and to achieve top management support. However, once the benefit of the tool was realized, commitment of top management was forthcoming. Resulting positive political climate enabled the execution of an organization-wide ISP in modeling the functions of the organization.

However, in the one instance where ISP was reportedly executed from inception, the interviewee reported that top management was committed, knowledgeable, and fully supportive of IE, IEF and the importance of ISP as the basis from which all future applications would follow. [Ref. 22] This type of enlightened leadership perceives the value of laying a solid bedrock of strategic planning prior to constructing major software applications. Therefore, all users interviewed agreed that support of upper level management is essential to success.

Major drawbacks to conducting an ISP are threefold: political, financial and educational. These considerations are intricately interconnected. The dilemma faced by IT professionals in this regard is that ISP is very expensive in both cost and non-cost issues. An ISP requires fiscal expenditures for man-months of both analysts and high ranking

user representatives, and training for both groups. Non-cost factors include overcoming political opposition, and securing universal agreement among top-level managers on the precise architecture of their corporate environment.

The political drawback concerns the commitment of top management to the tool and the ISP process. ISP is time consuming. All users admitted that six to eight months was typical for conducting an ISP, with an expected 18 month timeline anticipated for the NAMO project that involves the entire Navy's requirements. Furthermore, ISP requires cooperation and communication of all affected functional areas. Even with top management commitment, the potential exists for disputes or even sabotage of the ISP process when departments are asked to sacrifice or share data attributes to which they claim ownership. This can occur when attempting to modify data attributes to a unified format, such as the number of characters of a specific data element. However, top management is more easily persuaded to use the entire tool's capabilities, despite the cost, once credibility is established through successful pilot projects.

The financial factor affecting ISP also has repercussions on the political climate. Diverting highly paid individuals' time and effort to conduct an effective ISP can be a daunting up-front cost. While strategic planning efforts are underway, there is no product, nor visible progress toward a deliverable. As this lack of tangible results continues for

two or three fiscal quarters, allocation of money for personnel and resources assigned to this task requires steadfast management commitment. Beyond the actual drain on resources conducting the ISP comes another significant funding requirement - education/training.

In many cases, the choice of IEF requires a re-education of the MIS professional from conventional methodology on which they were weaned into the world of Information Engineering. If an organization is to do an effective ISP, even the non-MIS individuals in the organization who are assigned to the project must be trained in IE (at the very least). Such training is necessary to enable all personnel to speak the same language. This is required to create an effective and productive environment based on a shared understanding and communication. As previously mentioned, the learning curve for this tool is substantial and acquiring that knowledge without vendor offered training is unlikely to be productive, according to the users. Such IEF training is expensive. All users agreed that training is essential, for both project team members and for other personnel involved in using or contributing to the tool's optimal utilization.

These three drawbacks can be outweighed in DoD by compelling arguments concerning the need to promote both CIM and Total Quality Leadership (TQL). Lieutenant Commander Chase, USN, Project Management Officer at NAMO, indicated that

ISP fits very nicely with both philosophies being espoused by DoD today. The CIM initiative requires this kind of detailed strategic planning for information technology. NAMO's ISP was very effectively done using Total Quality Leadership's (TQL) Process Action Teams (PATs) to generate the product. This concept of operations for DoD agencies becomes ever more intriguing as both the CIM initiative and TQL broaden their impact on the developing future of IT in DoD.

2. Productivity Increases Cited

In cases cited by users, the productivity increase experienced through the use of IEF on projects that had been previously estimated or implemented using conventional methodology was noticeable. Completion was reported to take less than half the time and estimated man-years. This included the time required for training on the IEF tool. Subsequent increases were anticipated as MIS departments became better versed and more adept in the tool's utilization.

In 1982, the Federal Reserve Bank in St. Louis, Missouri, estimated a project for implementation using conventional methods as requiring 76 Man-Months (14 calendar months). This project had been attempted on three occasions using conventional methodology, each halting with the decision that the organization could not afford to implement it based

on its estimated costs. In 1988, it was completed using IEF in only 32 MM (eight calendar months).³

Another project, cited by one of the larger organizations surveyed, was originally developed in 22 person-years using conventional methods. Over the system's lifecycle, patchwork modifications had degraded system performance and maintenance efforts. The major upgrades required to revitalize the program led to the decision to replace the system. The resultant effort, using IEF, took fewer than 10 man-years. These reports seem to substantiate vendor claims of 50% effort reduction.

3. Type of Projects

All of the projects reported by users involved data management from one or more central data repositories. No reports were made concerning the use of IEF for real-time systems or embedded software. The data and process orientation of the tool lends itself more to data management applications.

4. Plaudits

Available technical support by TI received extremely high marks across the board. TI's staff seemed to extend themselves to ensure the success of each and every project venture of their clients. Intense competition for I_CASE

³ As reported by Mr. Jim Albenesius, Technical Project Coordinator, Federal Reserve Bank, St. Louis, MO.

market shares over the next three to five years will dictate survival or extinction for current I_CASE vendors. This pressure for user loyalty may explain the drive behind such personal attention.

Code generation is the major recognized power of the tool. It will literally produce 100%, error-free software (vendor claimed and substantiated by interviewed users). Users reported that they do indeed "throw away the source code" and as modifications occur, new programs are produced as required. The chief advantage, as cited by Mr. Albenesius, Technical Project Coordinator, Federal Reserve Bank, St. Louis, Missouri is that: "I don't have emergency maintenance on tool generated code." Emergency maintenance is that maintenance required when a system crashes due to critical errors. The ability to throw away source code lies in the fact that modifications are NEVER made to code in an IEF developed application. Changes and modifications are made at the appropriate design level and the entire application is regenerated to include the changes. [Ref. 23]

Mr. Albenesius reported an instance wherein a version of an application was almost fully developed except for some pending changes that were wholly dependent on results of a forthcoming vote by Congress. Within 24 hours of the vote, the bank was able to generate and distribute a functional, error-free application to all its field units. [Ref. 23] This is a stellar example of technology providing competitive

advantage. Organizations that rely on rapidly changing application requirements, and that fail to embrace some form of I_CASE technology, may find it difficult to remain competitive.

5. Warnings/Complaints

The chief complaint echoed by virtually all users was the incompatibility of the IEF tool with any other vendors' tools. The degree of portability is extremely low. This perception of IEF corroborates the findings in the *Computerworld* article, with its portability of other vendors' output being the most telling deficiency that required attention. [Ref. 18] However, TI will gladly provide at extra cost the technical support required to port one vendor's product into IEF.

The learning curve is extremely steep and training of core team members is essential. Most of the users indicated that both users for whom the application was being developed and higher level management could benefit from indoctrination into the basic conceptual framework of Information Engineering. Such training made development much more productive and efficient. Additionally, if functional area managers can be persuaded to invest the time and money to send at least one or two individuals with decision making capability to the BAA I or ISP vendor training, then the initial critical modeling work is greatly simplified. This

approach enables both client and analysts to be capable of discussing system requirements from a common methodological viewpoint.

Most users found that the introduction of IE required a cultural change from top management down since applications were no longer isolated, but part of a larger informational strategic plan. Additionally, analysts, designers, and programmers trained in conventional methodological approaches had to be re-educated to use IEF effectively. Although this change might appear to be the least challenging, one manager surveyed indicated that the "old timers" who were refusing to change were being placed in jobs maintaining old systems.

B. IEF DEFICIENCIES NOTED DURING DEVELOPMENT

1. Report Generation

IEF provides various reports of system development but does not provide a report generating capability in the final delivered application. Such requirements for accessing centrally stored information are meant to be satisfied utilizing any variety of report generating products available on the market. "The IEF currently provides no direct support for the creation of reports." [Ref. 5:p. 343]

2. Graphical User Interface (GUI)

IEF does support the designing of input screens. However, GUI's were designed to look as if they were emulating an IBM 3270. It does not provide a point-and-click capability

and actions are entered with function keys or command codes. It allows the arrangement of fields on data entry screens to closely resemble a source document, if one is used for data collection. Designers can set standards for colors, highlighting, and general format so that the look and feel of all components within the system are uniform. They have design options including use of multiple menus to simplify navigation among procedures or enabling the use of function keys and short command synonyms to provide quick access to the system's procedures. TI is working on improved GUIs as a separate utility for various operating system platforms. [Ref. 24]

3. Limited Languages Developed

The languages presently available for IEF code generation are limited to COBOL and C. This restriction significantly reduces the attractiveness of using IEF for code generation (construction toolset) at NPS. [Ref. 25] Public Law has mandated that, where cost effective, all DoD software shall be written in the programming language ADA.⁴ Future releases of IEF may support additional languages⁵ that might entice a change in planned utilization of IEF.

⁴ US Congress Department of Defense Appropriations Act 1991. Public Law 101-511 (November 5, 1990), 104STAT.1856-1914.

⁵ TI disclosed, in a 11 July 1991 press release, its plans to develop an ADA code generation capability for IEF by June 1992.

4. Training/Learning Curve

The steep learning curve for both converting to the IE methodology and training to use IEF necessitated a narrowing of scope and focus for the development of this phase of USAD. The amount of effort required for this education limited the accessible range of options that could be successfully accomplished given the level of experience and available time of the project development team.

The information in this chapter provided the infrastructure for the data and activity models created using IEF. These models were based on the Curricular Officers' requirements and necessary modifications are discussed in the next chapter.

VI. ANALYSIS OF CURRICULAR OFFICER REQUIREMENTS

A. MODIFICATION TO EXISTING REQUIREMENTS ANALYSIS

We did not conduct a preliminary investigation of Curricular Officers' requirements because of concerns that we might inadvertently rekindle user expectations that a new USAD system was imminent. We relied instead on a preliminary requirements analysis done last year to serve surrogate [Ref. 7]. Modifications to this analysis were required due to differences in representing data as entities in IEF (using the Information Engineering methodology) instead of as objects in the traditional Systems Design and Analysis approach. Clarifications were obtained [Ref. 12] to inconsistencies identified during an extensive analysis of Reference 7. The following adjustments were made:

1. Multi-valued (MV) objects/entities of the study have been modified to satisfy the following IEF requirement: "The IEF does not provide for multi-valued attributes, nor should it. Rather, one should remove an apparently multi-valued attribute to its own entity type and relate it to the original entity type via a 1:M [one to many] relationship." [Ref. 5:p. 160]
2. The study identifies the Thesis object/entity within the Student object/entity. However, a thesis could be jointly written by two students and therefore require representation as a separate entity. Likewise, Advisors and Second Readers maintain a one to one [1:1] relationship with a thesis and therefore are included within the Thesis entity.

3. IEF supports attribute names up to 32 characters in length. Titles or field names of data elements indicated in the study were therefore expanded to enhance readability and clarity.
4. Curriculum was identified as an attribute of the Student object/entity. Since a Curriculum maintains a 1:M relationship to the Student entity, it is represented as its own entity type.
5. To ensure that Curricular Officers access only records pertaining to their curricular office, passwords were added as additional security. A menu-driven, password-protected application could be designed to determine designated users access to specific areas of the database. This requirement fulfills MIS Department's concerns over protecting sections of the database from those who require/are permitted read-only access and have no need to write or modify such data elements. Use of passwords may also act as a verification of the active user profile that will provide access to that part of the database. For example, Curricular Officers should be able to view grades of their assigned students, but only the Registrar can Add/Modify grades. Additionally, Curricular Officers have expressed concern that they maintain unique access to their students' personal data.
6. Within the Grade Point Average object/entity, a 1:1 relationship exists between a student and their "overall" Quality Point Rating (QPR). Likewise, the Medical object/entity possesses the same 1:1 relationship. Both objects were therefore included in the Student entity.
7. The grade attribute was included in the Course object/entity. However, the grade attribute can only be associated with a particular student who completed a specific course, it was therefore placed in the composite object identified by the Student Course of Study entity type.
8. Curricular Officers requested the capability of using a template of a typical course of study for a particular curriculum based upon the type of student and refresher course requirements of the incoming student. This template consists of Multi-Valued (MV) attributes within MV attributes. Additionally, since a curriculum can recommend more than one typical course of study, three separate entity types had to be enumerated: the typical course of study, quarters of that typical course of study, and the courses associated with those quarters.

9. Book Claims and the Naval Book Eligibility entities had to be modeled separately. This was required to represent their MV attributes and to provide the ability for users to modify the total amount that all Naval students were eligible to claim without being forced to make changes to the program code.
10. Subtypes possess unique attributes in addition to those attributes inherited from their parent or supertypes. [Ref. 5:pp. 150, 156] This capability was used to represent a requested, scheduled, or completed course of a particular student as subtypes of the Student Course of Study entity (the supertype).

B. DEVELOPMENT OF CURRICULAR OFFICER REQUIREMENTS MODEL

The Appendices provide reports of data and activity models constructed in the BAA level of the IEF Planning Toolset [Ref. 26] and the IEF Analysis Toolset [Ref. 27] of the Curricular Officers' requirements for a recommended USAD system.

Appendix A displays an overall Entity Relationship Diagram (ERD) of the student information system. This diagram is used to depict relationships between entity types and to identify characteristics of those entity types. Two ERDs are presented. The first is a top level diagram which does not display partitions of three entity types: Student, Dependent, and Student Course of Study. It is evident that these entities are partitioned by three small circles in the upper right hand corner of the entity rectangles. The second is an expanded ERD indicating subtypes of the three entity types which were partitioned.

Appendix B provides an Entity Definition Report which contains information about the entity types and subtypes that are specified. Definitions include such information as entity type name, any aliases, description, properties, attributes and their aliases, relationships, and identifiers, if any. If an entity type has subtypes, each subtype is also defined. [Ref. 26:p. 20-7]

Appendix C supplies an Entity Hierarchy Report which contains information about the parent entity types in the model and their subtypes, including the identification of its attributes. [Ref. 26:p. 20-8]

Appendix D provides the Attribute Cross Reference Report which lists all attributes within the model alphabetically, to include IEF-supplied attributes. It lists each attribute name, associated entity type or subtype, and properties. [Ref. 27:p. 32-8]

Appendix E furnishes the Attribute Definition Report. This report contains information about the attributes that are specified in the data model, such as attribute name, and aliases, entity type, description, properties, length, default value, permitted values, and permitted values description. [Ref. 27:p. 32-9]

Appendix F presents the Process Definition (or Activity Definition) Report, which contains information about functions and processes of the activity model. [Ref. 27:p. 32-10]

Appendix G exhibits the Process (or Activity) Hierarchy Report which shows the hierarchy of activities in the activity model. [Ref. 27:p. 32-11]

Appendix H illustrates Action Diagrams of all processes developed in the activity model. A process is a defined business activity subordinate to a function or higher level process. It deals with what a business does, not how it is done. They have inputs which are used to perform work to produce outputs (inputs/outputs are called information views). [Ref. 28:p. 8-26] A process is a part of a function (an on-going, broad business activity) and deals with what a business does in particular. Its executions may be identified in terms of input and output of specific entities or data about specific entities. [Ref. 27:p. 12-3]

Each attribute within a developed data model must have a process which creates, updates, or deletes it (unless unique to the business function). Because of this requirement, various processes were developed over which the Curricular Officers' Business Area does not have control, but nevertheless must be given access. These processes are described as "System Gen" in the process descriptions of the Action Diagrams.

Appendix I portrays the Action Diagrams of action blocks developed in the activity model. These describe the logic of the algorithms used to derive a calculated data element.

Appendix J displays the Process Dependency Diagrams of the created activity model. These diagrams illustrate the sequence of events and flow of data required for each activity.

These data and activity models complete the Business Area Analysis (BAA) phase of IEF. The next phase involves fashioning details of users' interfaces with the developed system application. This process is the Business System Design (BSD) phase which could be the subject of future research, if NPS elects to continue this IEF based development of USAD. Present modeling was sufficient to generate recommendations concerning the utilization of IEF at NPS.

VII. CONCLUSIONS/RECOMMENDATIONS

Texas Instruments admits that IEF may be utilized commencing with the BAA, and bypassing the top-level ISP. This research shows such potential does exist. However, specific conditions must be present to recommend using IEF in this manner. Development of USAD is not such a project.

A. BASED ON INTERVIEW

An ISP is recommended since the Curricular Officer requirements cross functional boundaries with both the Registrar and Admissions Office requirements. The intertwined processes of handling student data require strategic planning that is the conventional starting point for the IEF. Regardless of the tool, an ISP (i.e. higher level planning phase) is recommended for production of the Curriculum Officer requirements specification.

All users interviewed agreed that an ISP is not necessary in specific cases. In many instances this decision is warranted since the costs and essential requirement of higher level management commitment to conduct the ISP may be lacking. An ISP can be skipped if, and only if, the business area chosen for the BAA starting point has requirements and processes that are well bounded and constrained.

The Curricular Officer business area routinely crosses boundaries in utilizing/supplying data from/to both the Registrar's and Admission Offices' business areas. An ISP would therefore be the appropriate first step. It would enhance the performance of the team tasked with development of the USAD.

A deliberate and meticulous ISP clearly defines functional responsibilities. Therefore, communication overhead between analysts/programmers and analysts/users is reduced and the rework necessary, due to changing user requirements, is mitigated because of up-front concurrence of top-level planners. Both maintenance of existing software and future software development applications at NPS would benefit since modifications are easily made to a documented strategic blueprint using IEF.

The sole suggestion from other IEF users that might obviate the need for an ISP at NPS, is whether an individual exists within the organization with superior knowledge and insight into the workings of the business. Such an individual can provide IEF analysts with the organizational overview and corporate policy decisions that would yield intimate knowledge of dynamic interrelationships between various offices involved with handling student data. If such an individual does not exist or the organizational political climate is too uncertain for such knowledge to reside in any one person, then an ISP is recommended.

If USAD is intended to satisfy some long-range requirement within the overall strategic plan of NPS, then an ISP is strongly recommended. While the size of USAD may not immediately dictate a requirement for an ISP, the scope of intended utility of IEF within NPS' corporate framework may present sufficient justification. This is especially pertinent in light of DoD's dedication to both the CIM initiative and TQL. Documenting NPS' mission and organizational objectives, and determining specific functions and responsibilities throughout the organization, facilitates the adoption of these corporate philosophies. NPS' commitment to a top-level ISP, and subsequent lower-level Departmental ISP's, based on this higher corporate strategy, is recommended to ensure a unified orientation for the entire organization.

While the current conduct of academic business can be modeled commencing with the BAA, it is suggested that this shortcut be avoided. Full capability of the tool purchased by the school can best be utilized for the long-range benefits inherent in comprehensive strategic planning.

B. BASED ON EXPERIENCE

In attempting to use the IEF tool to model the Curricular Officers' requirements, we discovered that the Curricular Officers' reliance on data from both the Registrar's and Admission's Offices precluded a simple modeling of only these requirements. In order to provide consistency checks through

the IEF tool, we had to include those entities that fell within the purview of the Registrar's or Admission's Offices. Without the availability of the higher level analysis that would have been conducted via the tool's ISP, the modeling of these areas was done from a logical extrapolation of known processes, rather than an internalized, documented strategic plan upon which work within the BAA could be based.

"Business areas are defined during ISP as pieces of the enterprise which can be analyzed independent of one another." Since the Curricular Officers' business area is not *independent* of the Registrar's or Admission's Offices, an ISP by definition is required in order to properly use IEF to model the Curricular Officers' business area requirements.

Additionally, experience gained in modeling the Curricular Officers' business area showed that an ISP would be extremely useful since "the scope of a business area should be analyzed all at once...to avoid scope creep and future system integration problems." [Ref. 13:p. 6]

C. BASED ON TI'S IEF PERSONNEL

Although the vendor's publications indicate the ability to launch a project at the BAA level, conversations with TI's technical experts indicate that an ISP is strongly recommended from the very beginning. [Ref. 19]

Long-term investment of time, personnel, and resources cause private industry to avoid conducting an ISP, according

to trends highlighted in surveys executed by TI. However, this is an initial shortcut used by IT managers to enable rapid production of a smaller application. A quick victory provides upper level management with evidence of the tools potential. Once credibility has been established, the track record seems to indicate that higher level management is then willing to invest in ISP and development of corporate-wide applications based on that ISP. [Ref. 19]

D. TRAINING (IE & IEF)

1. Based on Interviews

The single most important element stressed by all IEF users surveyed was the critical need for extensive training. The cost is high, but the learning curve is so steep otherwise that significant man-months will be wasted discovering intricacies of the toolset's capabilities. All users contended that long-range commitment to using IEF demanded the expenditure for training. While training costs contribute to the long break-even point of I_CASE, the critical nature of the requirement cannot be ignored. [Ref. 17]

We recommend that the project team assigned to use the tool and the Director, MIS be scheduled to take the BAA I, BAA II, and Business System Design (BSD) training. If one user from each of the functional areas can take the BAA I training, this will greatly facilitate the modeling process.

At a minimum, all users who will be involved with USAD system modeling and all the MIS staff should be thoroughly indoctrinated in IE. However, the limited use of the tool at NPS may not require a total commitment to IE - this would depend on the number of projects upon which IEF is expected to be used. Professor Kamel in the Information Systems group from Administrative Sciences is contemplating the initiation of an IE course which could be an excellent source of training for NPS MIS personnel.

If a true ISP of NPS is being considered, then an overview IE orientation training class is suggested for higher level managers as well. The language and concepts being discussed must have a firm footing in shared territory. Mr. Joe Joseph at NAMO is an excellent point of contact for this type of anecdotal experience and training. He is the Computer System Analyst primarily responsible for the conduct of ISP at NAMO. He suggested that, except for top management support, training is the most essential element for a successful ISP. He has conducted training classes for NAMO personnel in IE and ISP, and is an excellent source of expertise.⁶

2. Based on Experience

Without the training received via the tutorial, and especially training in BAA I provided by TI, modeling of the

⁶ For further information contact Mr. Joe Joseph at Autovon: 326-7900.

Curricular Officers' requirements using IEF would have been an order of magnitude more difficult. Formal training in both IE and IEF is strongly recommended.

The technical support provided by Ms. Jeffrie Penrod, of TI's Santa Clara office was superb. Willingly devoting many hours to application specific questions and making several "house calls" of many hours duration to assist in proper tool utilization made the learning curve a surmountable obstacle. Without her commitment to meeting user needs, professional concern, and teaching capabilities, our results would have been achieved only after significant wasted effort and time. We recommend the MIS Department maintain such a liaison during the development of USAD.

The learning curve would undoubtedly be more manageable for analysts who are trained in IE and who can devote the majority of their working hours to becoming competent with the tool, but as Mr. Joe Joseph pointed out: "We are all novices on the first day!"

E. IE METHODOLOGY AS COURSE AT NPS

For students in the IS curriculum to graduate with no knowledge of IE verges on negligence. The changing methodology of analysis and design, and growing acceptance of IE as a viable alternative to standard methodologies requires academic consideration of this subject for inclusion in a course.

We recommend IEF be used as the Analysis and Design tool for such a course. There also exists potential for productivity benefits to the school of having students doing work in the same toolset as the MIS department.

TI representatives have already indicated a willingness to support such a course with technical assistance and suggested course material for instructing IE (the co-developer of IE - James Martin - was primary consultant to TI for the years during the development and production of IEF).

F. RESTRICTED ACCESS

Required access restriction of various entity attributes can be designed in procedure diagrams in IEF which provide logic to describe how data is accessed. Password access should be used to further ensure that users are restricted in accessing/utilizing data elements not under their cognizance, in accordance with the Dean of Computers and Information guidance. The conduct of a thorough ISP will ensure these requirements are properly developed and organizationally acceptable (see H. below - the reasoning remains valid).

G. ACCOUNTABILITY/OWNERSHIP OF DATA ELEMENTS

One of the vital results of a thorough ISP is that all these types of issues/questions are answered by the appropriate level of managerial decision makers. By bringing key players into the process at the highest level of strategic

planning, agreements and decisions reached during an ISP ensure that resulting analysis and design will receive endorsement and support at all levels of management. The political infighting and disagreements that often accompany accountability/ownership become moot after a proper ISP has been conducted.

H. LIMITING IEF'S POTENTIAL

When TI's IEF can generate source code in a language approved for use at NPS, it is recommended that the school purchase the remaining toolsets to enable application construction via IEF. Utilizing an I_CASE tool for its front-end only defeats the purpose of such a tool since "most I_CASE solutions must be implemented in their entirety to begin receiving measurable benefits, thus increasing the initial investment outlay." [Ref. 17]

This type of comprehensive utilization of the tool would more readily justify the expenditures of resources devoted to properly modeling the NPS environment on the Planning, Analysis and Design toolsets already purchased.

However, IEF is **not recommended** for use on an isolated application such as the USAD project. Long-range (three years or greater) return on investment for I_CASE occurs only when organizational commitment is made for long-term, maximized productivity gains. [Ref. 17] In our opinion, documented productivity enhancements of I_CASE, and specifically IEF,

would not be realized at NPS without 1) utilizing the entire toolset for USAD's planning, production and maintenance and, 2) organizational commitment to utilization of I_CASE on future portfolio applications.

Additionally, we feel that IEF's extensive potential can be wholly realized at NPS only via a comprehensive I_CASE implementation strategy. We recommend that such a bold scheme be considered and a feasibility study commissioned.

APPENDIX A

The Entity Relationship Diagrams on the following pages depict entity types, their relationships, and certain aspects of their important properties. This data model provides "a comprehensive representation of the fundamental things of relevance to the business (entities types) and their interrelationships." [Ref. 28]

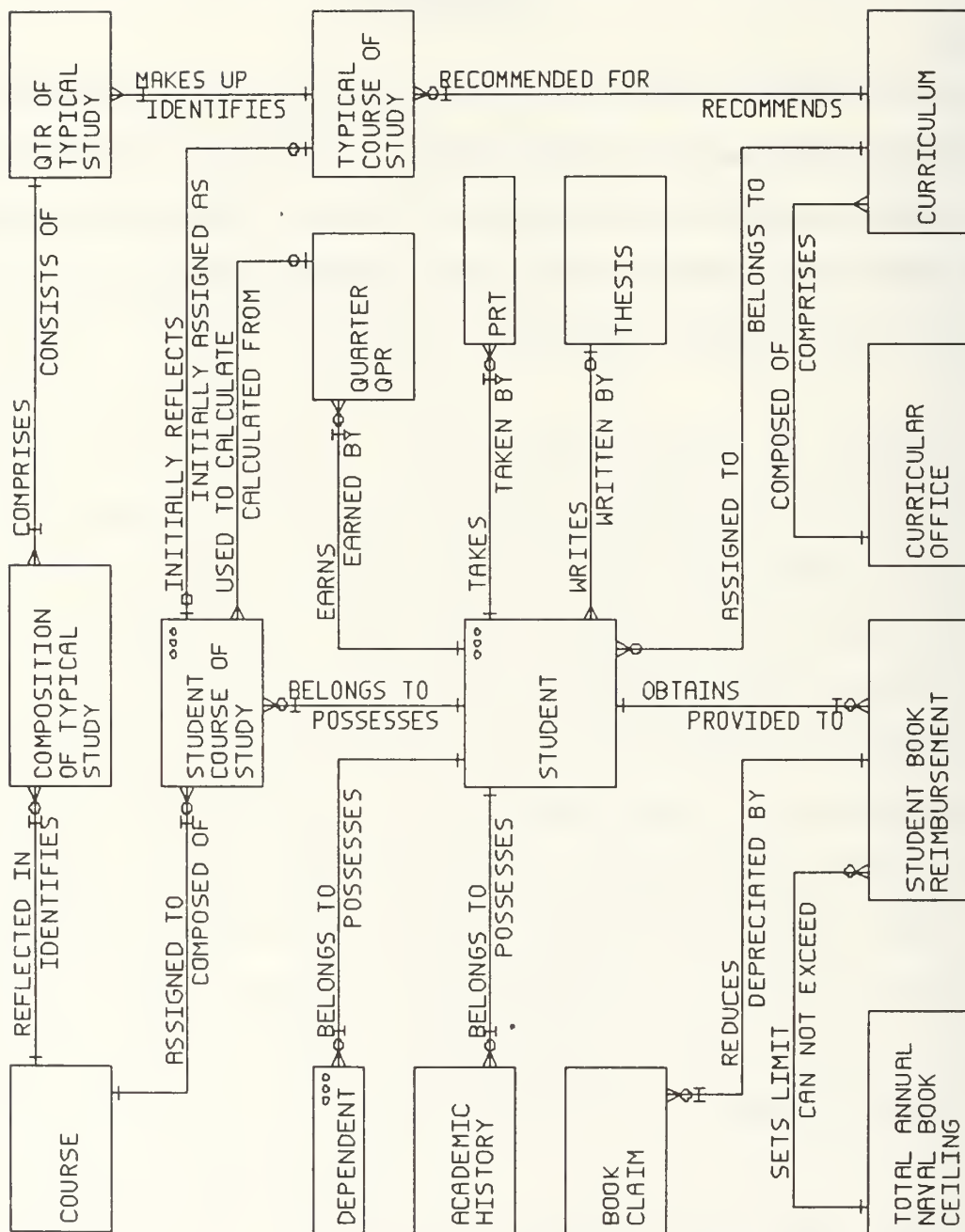
Rectangles represent entity types (known as objects in other fundamental design techniques), which are fundamental things of relevance to the business about which data is kept. Those rectangles with three small circles in the upper left hand corner indicate contracted entities (the entity is partitioned or subdivided based upon a classifying attribute).

Lines drawn between entity types depict relationships. The connection of these lines to rectangles determines the cardinality of the relationship. Cardinality is the number of times an entity occurrence can participate in a relationship. Those connection points which expand into a crow's foot connection indicate a multiple occurrence. Relationships are one to one, or one to many.

Perpendicular tick-marks on relationship lines indicate mandatory relationships while small circles indicate optional relationships.

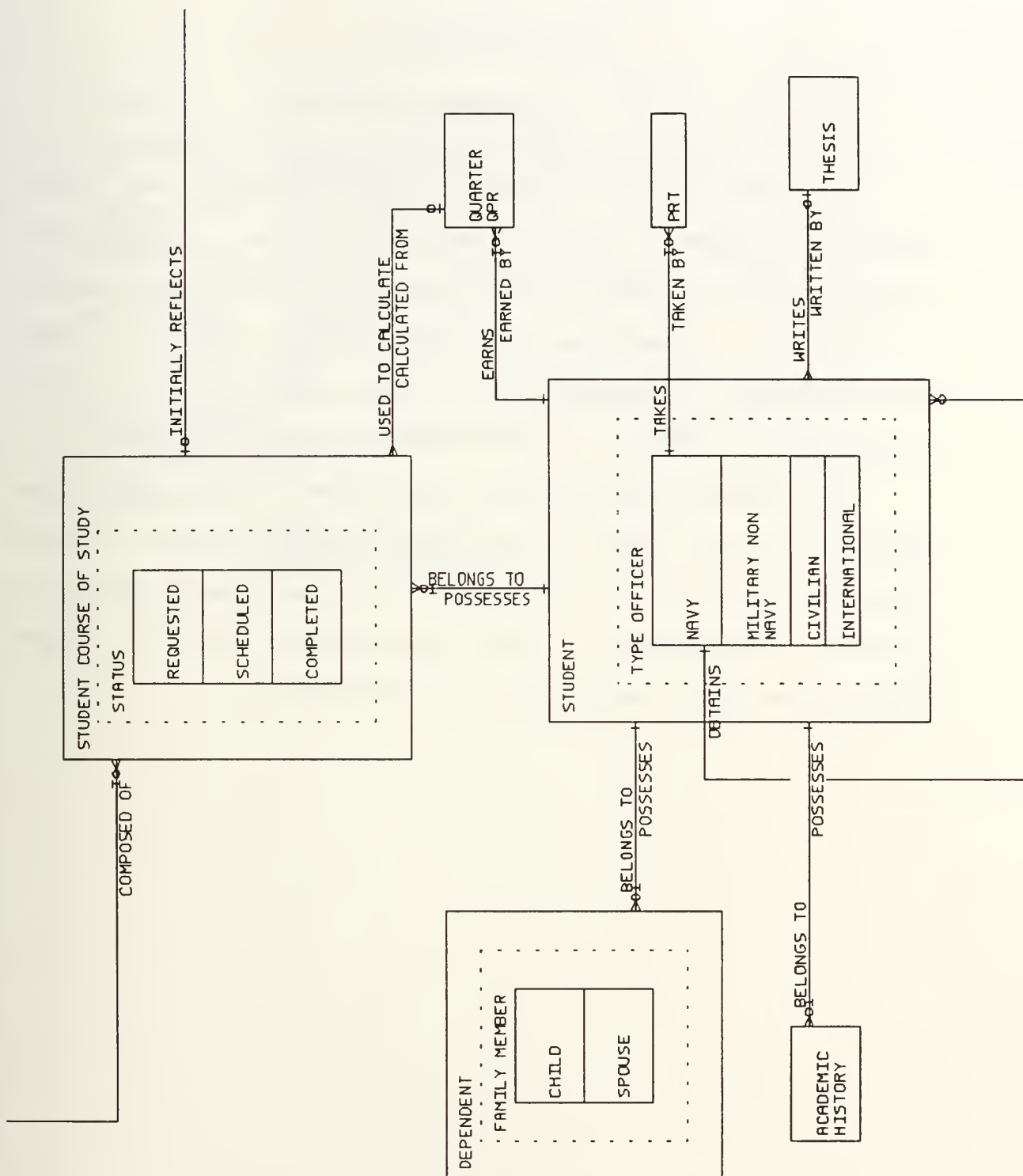
The small "I" on these relationship lines indicate that identifying attributes are obtained from the associated entity type.

The first Entity Relationship Diagram represents the overall data model of the student information system. The second provides an expansion of the three entity types; Student, Dependent, and Student Course of Study.



Curricular Officers' Student Academic Database System (CSADS)
Entity Relationship Diagram

Expanded CSADS Entity Relationship Diagram



APPENDIX B

The report on the following pages presents the Entity Definition segment of the designed system. Definitions include such information as the entity type name, any aliases, description, properties, attributes and their aliases, relationships, and identifiers, if any. If an entity type has subtypes, each subtype is also defined. [Ref. 26:p. 20-7]

The minimum and maximum occurrences in the properties portion of the report were subjectively imposed and are included primarily to allow the tool's consistency checks to execute successfully. Such information enables system administrators to estimate size requirements.

Entity Definition

Entity: ACADEMIC_HISTORY

Description: This identifies a particular student's Academic background prior to arrival at NPS.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 500 Avg Occ: 1000
Max Occ: 2000 Growth Rate: 5% per year

Attributes: SCHOOL
DEGREE
MAJOR
GPA
DATE

Relationships:
Always BELONGS_TO one STUDENT
cannot transfer.

Identifiers:
1 MAJOR
1 DEGREE
1 BELONGS_TO STUDENT

Entity Definition

Entity: BOOK_CLAIM

Description: This identifies the amount of a reimbursable claim for a specific student for a particular year (student book money); to be used as deduction from total remaining funds.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 1400 Avg Occ: 3200
Max Occ: 6400 Growth Rate: 5% per year

Attributes: AMOUNT_OF_CLAIM
ACADEMIC_QUARTER

Relationships:

Always REDUCES one STUDENT_BOOK_REIMBURSEMENT
cannot transfer.

Identifiers:

- 1 ACADEMIC_QUARTER
- 1 REDUCES STUDENT_BOOK_REIMBURSEMENT

Entity Definition

Entity: COMPOSITION_OF_TYPICAL_STUDY

Description: This links a specific course with a specified year & quarter of Typical Course of Study; presents the standardized template(s) as depicted in the course catalogue for the different curricula

Subject area: CURRICULAR_SADS

Properties: Min Occ: 100 Avg Occ: 1800
Max Occ: 2000 Growth Rate: 5% per year

Attributes: TYPE_OF_COURSE

Relationships:

Always COMPRISES one QTR_OF_TYPICAL_STUDY
cannot transfer.

Always IDENTIFIES one COURSE
can transfer.

Identifiers:

1 COMPRISES QTR_OF_TYPICAL_STUDY

1 IDENTIFIES COURSE

Entity Definition

Entity: COURSE

Description: This describes the entire list of available courses at the NPS based on course catalogue entries (under purview of Registrar - could be modeled here as an external object)

Subject area: CURRICULAR_SADS

Properties: Min Occ: 800 Avg Occ: 900
Max Occ: 1000 Growth Rate: 5% per year

Attributes: NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
ACADEMIC_DEPARTMENT_CODE
NUMBER

Relationships:

Sometimes (90%) REFLECTED_IN many COMPOSITION_OF_TYPICAL_STUDY
Cardinality Min: 1 (est) Max: 100 (est) Avg: 50
cannot transfer.

Sometimes (70%) ASSIGNED_TO many STUDENT_COURSE_OF_STUDY
Cardinality Min: 1 (est) Max: 2000 (est) Avg: 1700
cannot transfer.

Identifiers:

- 1 NUMBER
- 1 ACADEMIC_DEPARTMENT_CODE

Entity Definition

Entity: CURRICULAR_OFFICE

Description: This identifies the organizational entity responsible for the maintenance of the curricular programs.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 11 Avg Occ: 11
Max Occ: 20 Growth Rate: 1% per year

Attributes: TITLE
CODE
PASSWORD

Relationships:

Always COMPOSED_OF many CURRICULUM
Cardinality Min: 1 Max: 20 (est) Avg: 5
cannot transfer.

Identifiers:
1 CODE

Entity Definition

Entity: CURRICULUM

Description: This identifies the number & title of the curriculum program required for a specific degree

Subject area: CURRICULAR_SADS

Properties: Min Occ: 40 Avg Occ: 50
Max Occ: 60 Growth Rate: 5% per year

Attributes: TITLE
NUMBER

Relationships:

Sometimes (50%) BELONGS_TO many STUDENT

Cardinality Min: 1 (est) Max: 400 (est) Avg: 200
cannot transfer.

Sometimes (0%) RECOMMENDS many TYPICAL_COURSE_OF_STUDY

Cardinality Min: 1 (est) Max: 30 (est) Avg: 4
cannot transfer.

Always COMPRISES one CURRICULAR_OFFICE
cannot transfer.

Identifiers:

1 NUMBER

Entity Definition

Entity: DEPENDENT

Description: This identifies any known dependents (child/spouse) belonging to a particular student.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 500 Avg Occ: 2000
Max Occ: 4000 Growth Rate: 5% per year

Attributes: LAST_NAME
FIRST_NAME
FAMILY_MEMBER

Relationships:
Always BELONGS_TO one STUDENT
cannot transfer.

Identifiers:
1 LAST_NAME
1 FIRST_NAME
1 BELONGS_TO STUDENT

Partitioned by: FAMILY_MEMBER

Classifying Value	Subtype
-----	-----
SPOUSE	SPOUSE
CHILD	CHILD

Entity Definition

Entity: SPOUSE

Description: This identifies the dependent as a spouse and indicates if the individual is also enrolled at NPS.

Properties: Min Occ: 1000 Avg Occ: 1700
Max Occ: 2000 Growth Rate: 5% per year

Attributes: ALSO STUDENT

Subtype of: DEPENDENT

Inherited Attributes:

DEPENDENT:	LAST <u>NAME</u>
	FIRST <u>NAME</u>
	FAMILY <u>MEMBER</u>

Entity Definition

Entity: CHILD

Description: This identifies the dependent as a child and stores the birthdate and sex of the child.

Properties: Min Occ: 2000 Avg Occ: 6000
Max Occ: 8000 Growth Rate: 5% per year

Attributes: DATE_OF_BIRTH
GENDER

Subtype of: DEPENDENT

Inherited Attributes:
DEPENDENT: LAST_NAME
FIRST_NAME
FAMILY_MEMBER

Entity Definition

Entity: PRT

Description: This identifies the results of a Naval student's Physical Readiness Test.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 1000 Avg Occ: 2000
Max Occ: 4000 Growth Rate: 5% per year

Attributes: SCORE
BODY_FAT
DATE_OF_TEST

Relationships:
Always TAKEN_BY one NAVY
cannot transfer.

Identifiers:
1 DATE_OF_TEST
1 TAKEN_BY NAVY

Entity Definition

Entity: QTR_OF_TYPICAL_STUDY

Description: This entity reflects the quarter (numerical) of a particular curriculum's typical course of study. Each quarter is comprised of multiple courses.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 1 Avg Occ: 6
 Max Occ: 15 Growth Rate: 1% per year

Attributes: QUARTER_NUMBER

Relationships:

Always MAKES_UP one TYPICAL_COURSE_OF_STUDY
 cannot transfer.

Always CONSISTS_OF many COMPOSITION_OF_TYPICAL_STUDY
Cardinality Min: 1 (est) Max: 8 (est) Avg: 4
 cannot transfer.

Identifiers:

1 QUARTER_NUMBER
1 MAKES_UP TYPICAL_COURSE_OF_STUDY

Entity Definition

Entity: QUARTER_QPR

Description: This entity type identifies the Quality Point Ratings of a specific student for a particular quarter.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 200 Avg Occ: 16000
Max Occ: 24000 Growth Rate: 5% per year

Attributes: TOTAL
GRADUATE
ACADEMIC_YEAR
ACADEMIC_QUARTER

Relationships:

Always CALCULATED_FROM many STUDENT COURSE_OF_STUDY
Cardinality Min: 1 (est) Max: 12 (est) Avg: 6
cannot transfer.

Always EARNED_BY one STUDENT
cannot transfer.

Identifiers:

- 1 ACADEMIC_QUARTER
- 1 ACADEMIC_YEAR
- 1 EARNED_BY STUDENT

Entity Definition

Entity: STUDENT

Description: This identifies a projected/arrived/registered/graduated/dropped student at the Naval Postgraduate School.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 500 Avg Occ: 1000
Max Occ: 2000 Growth Rate: 5% per year

Attributes:

SSN	TYPE REFRESHER
PRESENT STATUS	TYPE OFFICER
LAST NAME	RECEIVED ORDERS TO ATTEND
FIRST NAME	MIDDLE INITIAL
SHORTNAME	GENDER
RANK	DATE OF RANK
PHONE NUMBER	STREET
CITY	ZIP CODE
SMC BOX NUMBER	LAMESA HOUSING OCCUPANT
SECTION NUMBER	SPLIT SECTION
PROPOSED NPS DEGREE	ACCREDITATION STATUS
NPS MAJOR	COMMISSIONING SOURCE
STUDY SPACE	LOCKER NUMBER
DATE OF BIRTH	PLACE OF BIRTH CITY
PLACE OF BIRTH STATE	DATE REPORTED ABOARD
SECURITY ACCESS	SECURITY BACKGROUND
LAST FITREP DATE	NEXT FITREP DUE
NEXT DUTY STATION	PREVIOUS DUTY STATION
DATE OF ORDERS	NAME OF SPONSOR
APC	DATE THAT A SPONSOR WAS ASSIGNED
DUAL DEGREE	DATE WELCOME PACKAGE SENT
CONVENING DATE	DATE SPONSOR LETTER SENT
MARITAL STATUS	IN BOUND STUDENT SPONSOR
LIBRARY CARD NUMBER	MAINFRAME ACCOUNT NUMBER
AIDS TEST DATE	PHYSICAL DATE
DENTAL DATE	ANTICIPATED GRADUATION DATE
TOTAL QPR	GRADUATE QPR
COMMENT1	STARTED PARENT CURRICULUM
COMPLETED FIRST REFRESHER QTR	

Relationships:

Sometimes (75%) EARNs many QUARTER_QPR

Cardinality Min: 1 (est) Max: 12 (est) Avg: 6
cannot transfer.

Sometimes (50%) POSSESSES many STUDENT_COURSE_OF_STUDY

Cardinality Min: 1 (est) Max: 80 (est) Avg: 50
cannot transfer.

Always ASSIGNED_TO one CURRICULUM

can transfer.

Sometimes (70%) POSSESSES many DEPENDENT

Cardinality Min: 1 (est) Max: 10 (est) Avg: 3
cannot transfer.

Sometimes (70%) WRITES one THESIS

can transfer.

Sometimes (90%) POSSESSES many ACADEMIC_HISTORY

Cardinality Min: 1 (est) Max: 4 (est) Avg: 1
cannot transfer.

Identifiers:

1 SSN

Partitioned by: TYPE_OFFICER

Classifying Value

Subtype

C

CIVILIAN

I

INTERNATIONAL

M

MILITARY_NON_NAVY

N

NAVY

Entity Definition

Entity: CIVILIAN

Description: This identifies the student as a civilian and indicates the type of program the individual will be enrolled in at NPS.

Properties: Min Occ: 1 Avg Occ: 50
Max Occ: 100 Growth Rate: 5% per year

Attributes: PROGRAM
Subtype of: STUDENT

Inherited Attributes:

STUDENT:

SSN	TYPE_REFRESHER
PRESENT STATUS	TYPE_OFFICER
LAST NAME	RECEIVED_ORDERS_TO_ATTEND
FIRST NAME	MIDDLE_INITIAL
SHORTNAME	GENDER
RANK	DATE_OF_RANK
PHONE NUMBER	STREET
CITY	ZIP_CODE
SMC_BOX_NUMBER	LAMESA_HOUSING_OCCUPANT
SECTION_NUMBER	SPLIT_SECTION
PROPOSED NPS_DEGREE	ACCREDITATION_STATUS
NPS MAJOR	COMMISSIONING_SOURCE
STUDY SPACE	LOCKER_NUMBER
DATE_OF_BIRTH	PLACE_OF_BIRTH_CITY
PLACE_OF_BIRTH STATE	DATE_REPORTED_ABOARD
SECURITY_ACCESS	SECURITY_BACKGROUND
LAST FITREP DATE	NEXT FITREP DUE
NEXT DUTY STATION	PREVIOUS DUTY STATION
DATE_OF_ORDERS	NAME_OF_SPONSOR
APC	DATE_THAT_A_SPONSOR_WAS_ASSIGNED
DUAL DEGREE	DATE_WELCOME_PACKAGE_SENT
CONVENING DATE	DATE_SPONSOR_LETTER_SENT
MARITAL STATUS	IN_BOUND_STUDENT_SPONSOR
LIBRARY_CARD NUMBER	MAINFRAME_ACCOUNT_NUMBER
AIDS TEST DATE	PHYSICAL DATE
DENTAL DATE	ANTICIPATED_GRADUATION_DATE
TOTAL_QPR	GRADUATE_QPR
COMMENT1	STARTED_PARENT_CURRICULUM
COMPLETED_FIRST_REFRESHER_QTR	

Entity Definition

Entity: INTERNATIONAL

Description: This identifies the student as an international student and indicates the country and military service of the individual.

Properties: Min Occ: 25 Avg Occ: 100
Max Occ: 300 Growth Rate: 5% per year

Attributes: INTERNATIONAL_SERVICE_COMPONENT
COUNTRY

Subtype of: STUDENT

Inherited Attributes:

STUDENT:	
SSN	TYPE_REFRESHER
PRESENT_STATUS	TYPE_OFFICER
LAST_NAME	RECEIVED_ORDERS_TO_ATTEND
FIRST_NAME	MIDDLE_INITIAL
SHORTNAME	GENDER
RANK	DATE_OF_RANK
PHONE_NUMBER	STREET
CITY	ZIP_CODE
SMC_BOX_NUMBER	LAMESA HOUSING_OCCUPANT
SECTION_NUMBER	SPLIT_SECTION
PROPOSED_NPS_DEGREE	ACCREDITATION_STATUS
NPS MAJOR	COMMISSIONING_SOURCE
STUDY SPACE	LOCKER NUMBER
DATE OF BIRTH	PLACE OF BIRTH CITY
PLACE OF BIRTH STATE	DATE REPORTED ABOARD
SECURITY_ACCESS	SECURITY_BACKGROUND
LAST FITREP DATE	NEXT FITREP DUE
NEXT DUTY STATION	PREVIOUS DUTY STATION
DATE OF ORDERS	NAME OF SPONSOR
APC	DATE THAT A SPONSOR WAS ASSIGNED
DUAL DEGREE	DATE WELCOME PACKAGE SENT
CONVENING DATE	DATE SPONSOR LETTER SENT
MARITAL STATUS	IN BOUND STUDENT SPONSOR
LIBRARY_CARD NUMBER	MAINFRAME ACCOUNT NUMBER
AIDS TEST DATE	PHYSICAL DATE
DENTAL DATE	ANTICIPATED GRADUATION DATE
TOTAL QPR	GRADUATE QPR
COMMENT1	STARTED_PARENT_CURRICULUM
COMPLETED_FIRST_REFRESHER_QTR	

Entity Definition

Entity: MILITARY_NON_NAVY

Description: This identifies a student as belonging to a US Military component other than the Navy. The individual's service component is identified.

Properties: Min Occ: 400 Avg Occ: 500
Max Occ: 1000 Growth Rate: 5% per year

Attributes: SERVICE
Subtype of: STUDENT

Inherited Attributes:

STUDENT:

SSN	TYPE_REFRESHER
PRESENT STATUS	TYPE_OFFICER
LAST NAME	RECEIVED_ORDERS_TO_ATTEND
FIRST NAME	MIDDLE_INITIAL
SHORTNAME	GENDER
RANK	DATE_OF_RANK
PHONE_NUMBER	STREET
CITY	ZIP_CODE
SMC_BOX_NUMBER	LAMESA_HOUSING_OCCUPANT
SECTION_NUMBER	SPLIT_SECTION
PROPOSED NPS_DEGREE	ACCREDITATION STATUS
NPS MAJOR	COMMISSIONING_SOURCE
STUDY SPACE	LOCKER_NUMBER
DATE OF BIRTH	PLACE OF BIRTH CITY
PLACE OF BIRTH STATE	DATE REPORTED ABOARD
SECURITY_ACCESS	SECURITY_BACKGROUND
LAST FITREP DATE	NEXT FITREP DUE
NEXT DUTY STATION	PREVIOUS DUTY STATION
DATE OF ORDERS	NAME OF SPONSOR
APC	DATE THAT A SPONSOR WAS ASSIGNED
DUAL DEGREE	DATE WELCOME PACKAGE SENT
CONVENING DATE	DATE SPONSOR LETTER SENT
MARITAL STATUS	IN BOUND STUDENT SPONSOR
LIBRARY_CARD_NUMBER	MAINFRAME ACCOUNT_NUMBER
AIDS TEST DATE	PHYSICAL DATE
DENTAL DATE	ANTICIPATED GRADUATION DATE
TOTAL QPR	GRADUATE QPR
COMMENT1	STARTED PARENT CURRICULUM
COMPLETED_FIRST_REFRESHER_QTR	

Entity Definition

Entity: NAVY

Description: This indicates that a student is a member of the US Navy, and identifies the officer's lineal number year group, and officer designator.

Properties: Min Occ: 400 Avg Occ: 800
Max Occ: 1600 Growth Rate: 5% per year

Attributes: OFFICER DESIGNATOR
LINEAL_NUMBER
OFFICER_YEAR_GROUP

Relationships:

Sometimes (50%) OBTAINS many STUDENT_BOOK_REIMBURSEMENT
Cardinality Min: 1 (est) Max: 3 (est) Avg: 1
cannot transfer.

Sometimes (80%) TAKES many PRT
Cardinality Min: 2 (est) Max: 4 (est) Avg: 2
cannot transfer.

Subtype of: STUDENT

Inherited Attributes:

STUDENT:

SSN	TYPE_REFRESHER
PRESENT STATUS	TYPE OFFICER
LAST NAME	RECEIVED ORDERS_TO_ATTEND
FIRST NAME	MIDDLE_INITIAL
SHORTNAME	GENDER
RANK	DATE_OF_RANK
PHONE_NUMBER	STREET
CITY	ZIP_CODE
SMC_BOX_NUMBER	LAMESA HOUSING_OCCUPANT
SECTION_NUMBER	SPLIT_SECTION
PROPOSED NPS_DEGREE	ACCREDITATION STATUS
NPS MAJOR	COMMISSIONING_SOURCE
STUDY SPACE	LOCKER NUMBER
DATE OF BIRTH	PLACE OF BIRTH CITY
PLACE OF BIRTH STATE	DATE REPORTED ABOARD
SECURITY ACCESS	SECURITY_BACKGROUND
LAST FITREP DATE	NEXT FITREP DUE
NEXT DUTY STATION	PREVIOUS DUTY STATION
DATE OF ORDERS	NAME OF SPONSOR
APC	DATE THAT A SPONSOR WAS ASSIGNED
DUAL DEGREE	DATE_WELCOME_PACKAGE_SENT
CONVENING DATE	DATE SPONSOR LETTER SENT
MARITAL STATUS	IN BOUND STUDENT SPONSOR
LIBRARY_CARD_NUMBER	MAINFRAME ACCOUNT_NUMBER
AIDS TEST DATE	PHYSICAL DATE
DENTAL DATE	ANTICIPATED GRADUATION_DATE
TOTAL_QPR	GRADUATE_QPR
COMMENT1	STARTED_PARENT_CURRICULUM
COMPLETED_FIRST_REFRESHER_QTR	

Entity Definition

Entity: STUDENT_BOOK_REIMBURSEMENT

Description: This reflects the available balance of book reimbursement for a particular student for a specific year; this amount cannot exceed the total Navy_Book_Eligibility ceiling imposed by NAVREGs.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 350 Avg Occ: 800
Max Occ: 1600 Growth Rate: 5% per year

Attributes: AMOUNT_REMAINING
TOTAL_AMOUNT_ELIGIBLE
NUMBER_ACADEMIC_QTRS_AUTHORIZED
YEAR

Relationships:

Always CAN NOT EXCEED one TOTAL_ANNUAL_NAVAL_BOOK_CEILING
cannot transfer.

Always PROVIDED_TO one NAVY
cannot transfer.

Sometimes (50%) DEPRECIATED_BY many BOOK_CLAIM
Cardinality Min: 1 (est) Max: 4 (est) Avg: 1
cannot transfer.

Identifiers:

1 YEAR
1 PROVIDED_TO NAVY

Entity Definition

Entity: STUDENT_COURSE_OF_STUDY

Description: This identifies all the courses that a student requests/schedules/completes at NPS; initially courses are requested using the Typical_Course_of_Study for a particular curriculum and is guided by the Courses or Offered_Courses available at the time of request; when scheduled, a section number is assigned and upon termination of student participation, a grade is earned.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 2000 Avg Occ: 4000
Max Occ: 8000 Growth Rate: 5% per year

Attributes: STATUS
VALIDATION
PASS_FAIL
ACADEMIC_YEAR
ACADEMIC_QUARTER

Relationships:
Sometimes (50%) USED_TO_CALCULATE one QUARTER_QPR
cannot transfer.
Sometimes (20%) INITIALLY_REFLECTS one TYPICAL_COURSE_OF_STUDY
cannot transfer.
Always BELONGS_TO one STUDENT
cannot transfer.
Always COMPOSED_OF one COURSE
can transfer.

Identifiers:
1 BELONGS_TO STUDENT
1 COMPOSED_OF COURSE

Partitioned by: STATUS

Classifying Value	Subtype
-----	-----
C	COMPLETED
S	SCHEDULED
R	REQUESTED

Entity Definition

Entity: COMPLETED

Description: This indicates that a course has been completed by a particular student. The student's grade is identified.

Properties: Min Occ: 2000 Avg Occ: 10000
Max Occ: 20000 Growth Rate: 5% per year

Attributes: GRADE

Subtype of: STUDENT_COURSE_OF_STUDY

Inherited Attributes:

STUDENT_COURSE_OF_STUDY:	STATUS
	VALIDATION
	PASS_FAIL
	ACADEMIC_YEAR
	ACADEMIC_QUARTER

Entity Definition

Entity: SCHEDULED

Description: This indicates that a requested course has been scheduled and identifies the section number of that particular course of which a student possesses enrollment.

Properties: Min Occ: 2000 Avg Occ: 10000
Max Occ: 20000 Growth Rate: 5% per year

Attributes: SECTION_NUMBER

Subtype of: STUDENT_COURSE_OF_STUDY

Inherited Attributes:

STUDENT_COURSE_OF_STUDY:	STATUS
	VALIDATION
	PASS_FAIL
	ACADEMIC_YEAR
	ACADEMIC_QUARTER

Entity Definition

Entity: REQUESTED

Description: This indicates that a course has been requested by a particular student.

Properties: Min Occ: 2000 Avg Occ: 10000
Max Occ: 20000 Growth Rate: 5% per year

Subtype of: STUDENT_COURSE_OF_STUDY

Inherited Attributes:

STUDENT_COURSE_OF_STUDY:	STATUS
	VALIDATION
	PASS_FAIL
	ACADEMIC_YEAR
	ACADEMIC_QUARTER

Entity Definition

Entity: THESIS

Description: This identifies the thesis written by one or more students.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 500 Avg Occ: 1000
Max Occ: 2000 Growth Rate: 5% per year

Attributes: NUMBER
DUE_DATE_MONTH
DUE_DATE_YEAR
TITLE
STATUS
ADVISOR
SECOND_READER
CLASSIFIED
JOINT

Relationships:
Always WRITTEN_BY many STUDENT
Cardinality Min: 1 Max: 2 (est) Avg: 1
cannot transfer.

Identifiers:
1 DUE_DATE_YEAR
1 NUMBER

Entity Definition

Entity: TOTAL_ANNUAL_NAVAL_BOOK_CEILING

Description: This is the funding ceiling allocated to all Naval students for book reimbursement during the year; used as basis for calculating a particular student's book money based on # of quarters remaining and previously submitted claims. No claims can be made if the figure here = \$0.00

Subject area: CURRICULAR_SADS

Properties: Min Occ: 1 Avg Occ: 1
Max Occ: 5 Growth Rate: 1% per year

Attributes: TOTAL_AMOUNT
DATE_IMPLEMENTED

Relationships:
Sometimes (50%) SETS_LIMIT many STUDENT_BOOK_REIMBURSEMENT
Cardinality Min: 400 (est) Max: 2000 (est) Avg: 1000
cannot transfer.

Identifiers:
1 DATE_IMPLEMENTED

Entity Definition

Entity: TYPICAL_COURSE_OF_STUDY

Description: This uniquely identifies (via Year & Quarter) a Matrix_of_Typical_Study for a particular curriculum; used once for every student (if at all) - once assigned to a specific student, is no longer used.

Subject area: CURRICULAR_SADS

Properties: Min Occ: 40 Avg Occ: 80
Max Occ: 120 Growth Rate: 5% per year

Attributes: TYPE_STUDENT
REFRESHER_REQUIREMENTS

Relationships:

Always IDENTIFIES many QTR_OF_TYPICAL_STUDY
Cardinality Min: 1 (est) Max: 12 (est) Avg: 6
cannot transfer.

Sometimes (50%) INITIALLY_ASSIGNED_AS one STUDENT_COURSE_OF_STUDY
cannot transfer.

Always RECOMMENDED_FOR one CURRICULUM
cannot transfer.

Identifiers:

- 1 REFRESHER_REQUIREMENTS
- 1 TYPE_STUDENT
- 1 RECOMMENDED_FOR CURRICULUM

-End of Report-

APPENDIX C

The report on the following pages, defines the Entity Hierarchy of the data model designed for the student information system which provides information about the parent entity types in the model and their subtypes, including identification of their attributes. [Ref. 26:p. 20-8]

Entity Hierarchy

Entity: ACADEMIC_HISTORY
Attrs: SCHOOL
DEGREE
MAJOR
GPA
DATE

Entity: BOOK_CLAIM
Attrs: AMOUNT_OF_CLAIM
ACADEMIC_QUARTER

Entity: COMPOSITION_OF_TYPICAL_STUDY
Attrs: TYPE_OF_COURSE

Entity: COURSE
Attrs: NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
ACADEMIC_DEPARTMENT_CODE
NUMBER

Entity: CURRICULAR_OFFICE
Attrs: TITLE
CODE
PASSWORD

Entity: CURRICULUM
Attrs: TITLE
NUMBER

Entity: DEPENDENT
Attrs: LAST_NAME
FIRST_NAME
FAMILY_MEMBER

Partng: FAMILY_MEMBER
Subtype: SPOUSE
Attrs: ALSO_STUDENT
Subtype: CHILD
Attrs: DATE_OF_BIRTH
GENDER

Entity: PRT
Attrs: SCORE
BODY_FAT
DATE_OF_TEST

Entity: QTR_OF_TYPICAL_STUDY
Attrs: QUARTER_NUMBER

Entity Hierarchy

Entity: QUARTER_QPR

Attrs: TOTAL
GRADUATE
ACADEMIC_YEAR
ACADEMIC_QUARTER

Entity: STUDENT

SSN	TYPE_REFRESHER
PRESENT_STATUS	RECEIVED_ORDERS_TO_ATTEND
LAST_NAME	FIRST_NAME
MIDDLE_INITIAL	SHORTNAME
GENDER	RANK
DATE_OF_RANK	PHONE_NUMBER
STREET	CITY
ZIP_CODE	LAMESA_HOUSING_OCCUPANT
SMC_BOX_NUMBER	DATE_REPORTED_ABOARD
CONVENING_DATE	ANTICIPATED_GRADUATION_DATE
SECTION_NUMBER	SPLIT_SECTION
PROPOSED_NPS_DEGREE	ACCREDITATION_STATUS
NPS_MAJOR	APC
STUDY_SPACE	LOCKER_NUMBER
MARITAL_STATUS	DATE_OF_BIRTH
PLACE_OF_BIRTH_CITY	PLACE_OF_BIRTH_STATE
SECURITY_BACKGROUND	SECURITY_ACCESS
LAST_FITREP_DATE	NEXT_FITREP_DUE
PREVIOUS_DUTY_STATION	NEXT_DUTY_STATION
DUAL_DEGREE	DATE_THAT_A_SPONSOR_WAS_ASSIGNED
NAME_OF_SPONSOR	DATE_WELCOME_PACKAGE_SENT
DATE_OF_ORDERS	DATE_SPONSOR_LETTER_SENT
COMMISSIONING_SOURCE	MAINFRAME_ACCOUNT_NUMBER
PHYSICAL_DATE	LIBRARY_CARD_NUMBER
DENTAL_DATE	IN_BOUND_STUDENT_SPONSOR
AIDS_TEST_DATE	COMPLETED_FIRST_REFRESHER_QTR
COMMENT1	STARTED_PARENT_CURRICULUM
TOTAL_QPR	GRADUATE_QPR
TYPE_OFFICER	

Partng:	TYPE_OFFICER
Subtype:	CIVILIAN
Attrs:	PROGRAM
Subtype:	INTERNATIONAL
Attrs:	INTERNATIONAL_SERVICE_COMPONENT COUNTRY
Subtype:	MILITARY_NON_NAVY
Attrs:	SERVICE
Subtype:	NAVY
Attrs:	OFFICER_DESIGNATOR LINEAL_NUMBER OFFICER_YEAR_GROUP

Entity Hierarchy

Entity: STUDENT_BOOK_REIMBURSEMENT
Attrs: AMOUNT_REMAINING
TOTAL_AMOUNT_ELIGIBLE
NUMBER_ACADEMIC_QTRS_AUTHORIZED
YEAR

Entity: STUDENT_COURSE_OF_STUDY
Attrs: VALIDATION
PASS_FAIL
ACADEMIC_YEAR
ACADEMIC_QUARTER
STATUS

Partng: STATUS
Subtype: COMPLETED
Attrs: GRADE
Subtype: SCHEDULED
Attrs: SECTION_NUMBER
Subtype: REQUESTED

Entity: THESIS
Attrs: NUMBER
DUE_DATE_MONTH
DUE_DATE_YEAR
TITLE
STATUS
ADVISOR
SECOND_READER
CLASSIFIED
JOINT

Entity: TOTAL_ANNUAL_NAVAL_BOOK_CEILING
Attrs: TOTAL_AMOUNT
DATE_IMPLEMENTED

Entity: TYPICAL_COURSE_OF_STUDY
Attrs: TYPE_STUDENT
REFRESHER_REQUIREMENTS

-End of Report-

APPENDIX D

The report on the following pages defines the Attribute Cross Reference of the designed system which lists all attributes within the model alphabetically, to include IEF-supplied attributes. It lists each attribute name, associated entity type or subtype, and properties. [Ref. 27:p. 32-8]

Attribute Cross Reference

attribute:	ACADEMIC_DEPARTMENT_CODE
entity:	COURSE
properties:	Mandatory Basic Text Length: 2
attribute:	ACADEMIC_QUARTER
entity:	QUARTER_QPR
properties:	Mandatory Basic Text Length: 4
attribute:	ACADEMIC_QUARTER
entity:	STUDENT_COURSE_OF_STUDY
properties:	Mandatory Basic Text Length: 4
attribute:	ACADEMIC_QUARTER
entity:	BOOK_CLAIM
properties:	Mandatory Basic Text Length: 4
attribute:	ACADEMIC_YEAR
entity:	STUDENT_COURSE_OF_STUDY
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	ACADEMIC_YEAR
entity:	QUARTER_QPR
properties:	Mandatory Basic Number Length: 2 Decimal: 0
attribute:	ACCREDITATION_STATUS
entity:	STUDENT
properties:	Optional Basic Text Length: 1
attribute:	ADVISOR
entity:	THESIS
properties:	Optional Basic Text Length: 22
attribute:	AIDS_TEST_DATE
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	ALSO_STUDENT
subtype:	SPOUSE of entity: DEPENDENT
properties:	Optional Basic Text Length: 1
attribute:	AMOUNT_OF_CLAIM
entity:	BOOK_CLAIM
properties:	Mandatory Basic Number Length: 5 Decimal: 2
attribute:	AMOUNT_REMAINING
entity:	STUDENT_BOOK_REIMBURSEMENT
properties:	Mandatory Derived Number Length: 6 Decimal: 2

Attribute Cross Reference

attribute:	ANTICIPATED_GRADUATION_DATE
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	APC
entity:	STUDENT
properties:	Optional Basic Number Length: 3 Decimal: 0
attribute:	AVERAGE_CURRENCY
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 11 Decimal: 2
attribute:	AVERAGE_INTEGER
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	AVERAGE_REAL
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 15 Decimal: 4
attribute:	BODY_FAT
entity:	PRT
properties:	Optional Basic Number Length: 2 Decimal: 0
attribute:	CITY
entity:	STUDENT
properties:	Optional Basic Text Length: 22
attribute:	CLASSIFIED
entity:	THESIS
properties:	Mandatory Basic Text Length: 1
attribute:	CODE
entity:	CURRICULAR_OFFICE
properties:	Mandatory Basic Text Length: 2
attribute:	COMMENT1
entity:	STUDENT
properties:	Optional Basic Text Length: 30
attribute:	COMMISSIONING_SOURCE
entity:	STUDENT
properties:	Optional Basic Text Length: 6
attribute:	COMPLETED_FIRST_REFRESHER_QTR
entity:	STUDENT
properties:	Mandatory Basic Text Length: 2

Attribute Cross Reference

attribute:	CONVENING_DATE
entity:	STUDENT
properties:	Mandatory Basic Date Length: 8
attribute:	COUNT
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	COUNTRY
subtype:	INTERNATIONAL of entity: STUDENT
properties:	Optional Designed Text Length: 10
attribute:	CUMULATIVE_HOURS
entity:	GAINED (local work element creation)
properties:	Mandatory Basic Number Length: 3 Decimal: 0
attribute:	CUMULATIVE_VALUE
entity:	POINT (local work element creation)
properties:	Mandatory Basic Number Length: 6 Decimal: 2
attribute:	CYYMMDD_NUMBER
entity:	NUMCYY (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	CYYMMDD_NUMBER
entity:	CYYNUM (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	CYYMMDD_NUMBER
entity:	CYYDATE (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	DATE
entity:	ACADEMIC_HISTORY
properties:	Optional Basic Date Length: 8
attribute:	DATE_IMPLEMENTED
entity:	TOTAL ANNUAL NAVAL BOOK CEILING
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_OF_BIRTH
subtype:	CHILD of entity: DEPENDENT
properties:	Optional Basic Date Length: 8
attribute:	DATE_OF_BIRTH
entity:	STUDENT
properties:	Optional Basic Date Length: 8

Attribute Cross Reference

attribute:	DATE_OF_ORDERS
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DATE_OF_RANK
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DATE_OF_TEST
entity:	PRT
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_REPORTED_ABOARD
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DATE_SPONSOR_LETTER_SENT
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DATE_STRING
entity:	DATEJUL (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	DATE_STRING
entity:	DATETEXT (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 10
attribute:	DATE_THAT_A_SPONSOR_WAS_ASSIGNED
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DATE_VALUE
entity:	NUMDATE (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DATENUM (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	JULDATE (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	YEAR (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8

Attribute Cross Reference

attribute:	DATE_VALUE
entity:	CYYDATE (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DAYOFWEEK (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DAY (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	MONTH (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DAYS (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DATEDAYS (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DATEJUL (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_VALUE
entity:	DATETEXT (available ief entity & attribute)
properties:	Mandatory Basic Date Length: 8
attribute:	DATE_WELCOME_PACKAGE_SENT
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DAY
entity:	DAY (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	DAY OF WEEK
entity:	DAYOFWEEK (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 9
attribute:	DEGREE
entity:	ACADEMIC_HISTORY
properties:	Mandatory Basic Text Length: 15

Attribute Cross Reference

attribute:	DENTAL_DATE
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	DUAL_DEGREE
entity:	STUDENT
properties:	Optional Basic Text Length: 25
attribute:	DUE_DATE_MONTH
entity:	THESIS
properties:	Mandatory Basic Text Length: 3
attribute:	DUE_DATE_YEAR
entity:	THESIS
properties:	Mandatory Basic Number Length: 2 Decimal: 0
attribute:	FAMILY_MEMBER
entity:	DEPENDENT
properties:	Mandatory Basic Text Length: 6
attribute:	FIRST_NAME
entity:	DEPENDENT
properties:	Mandatory Basic Text Length: 15
attribute:	FIRST_NAME
entity:	STUDENT
properties:	Mandatory Basic Text Length: 15
attribute:	FLAG
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Text Length: 1
attribute:	GENDER
subtype:	CHILD of entity: DEPENDENT
properties:	Optional Designed Text Length: 1
attribute:	GENDER
entity:	STUDENT
properties:	Optional Basic Text Length: 1
attribute:	GPA
entity:	ACADEMIC_HISTORY
properties:	Optional Basic Number Length: 3 Decimal: 2
attribute:	GRADE
subtype:	COMPLETED of entity: STUDENT_COURSE_OF_STUDY
properties:	Mandatory Designed Text Length: 2

Attribute Cross Reference

attribute:	GRADUATE
entity:	QUARTER_QPR
properties:	Mandatory Derived Number Length: 4 Decimal: 2
attribute:	GRADUATE_QPR
entity:	STUDENT
properties:	Optional Derived Number Length: 4 Decimal: 2
attribute:	GRADUATION_MONTH
entity:	DATE (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 2 Decimal: 0
attribute:	GRADUATION_YEAR
entity:	DATE (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	HOURL
entity:	HOURL (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	HOURS
entity:	GAINED (local work element creation)
properties:	Mandatory Basic Number Length: 1 Decimal: 0
attribute:	INCOMING_MONTH
entity:	DATE (local work element creation)
properties:	Mandatory Basic Number Length: 2 Decimal: 0
attribute:	INCOMING_YEAR
entity:	DATE (local work element creation)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	INDEX
entity:	FIND (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	INPUT_STRING
entity:	TRIM (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	INPUT_STRING
entity:	SUBSTR (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	INPUT_STRING
entity:	LENGTH (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096

Attribute Cross Reference

attribute:	INTERNATIONAL_SERVICE_COMPONENT
subtype:	INTERNATIONAL of entity: STUDENT
properties:	Mandatory Basic Text Length: 5
attribute:	IN_BOUND_STUDENT_SPONSOR
entity:	STUDENT
properties:	Optional Basic Text Length: 1
attribute:	JOINT
entity:	THESIS
properties:	Mandatory Basic Text Length: 1
attribute:	JULIAN_DATE
entity:	JULDATE (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	LAB_CREDIT_HOURS
entity:	COURSE
properties:	Mandatory Basic Number Length: 1 Decimal: 0
attribute:	LAMESA_HOUSING_OCCUPANT
entity:	STUDENT
properties:	Optional Basic Text Length: 1
attribute:	LAST_FITREP_DATE
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	LAST_NAME
entity:	DEPENDENT
properties:	Mandatory Basic Text Length: 23
attribute:	LAST_NAME
entity:	STUDENT
properties:	Mandatory Basic Text Length: 23
attribute:	LECTURE_CREDIT_HOURS
entity:	COURSE
properties:	Mandatory Basic Number Length: 1 Decimal: 0
attribute:	LIBRARY_CARD_NUMBER
entity:	STUDENT
properties:	Optional Basic Number Length: 7 Decimal: 0
attribute:	LINEAL_NUMBER
subtype:	NAVY of entity: STUDENT
properties:	Optional Basic Number Length: 8 Decimal: 0

Attribute Cross Reference

attribute:	LOCKER_NUMBER
entity:	STUDENT
properties:	Optional Basic Number Length: 3 Decimal: 0
attribute:	MAINFRAME_ACCOUNT_NUMBER
entity:	STUDENT
properties:	Optional Basic Text Length: 5
attribute:	MAJOR
entity:	ACADEMIC_HISTORY
properties:	Mandatory Basic Text Length: 15
attribute:	MARITAL_STATUS
entity:	STUDENT
properties:	Optional Basic Text Length: 1
attribute:	MIDDLE_INITIAL
entity:	STUDENT
properties:	Optional Basic Text Length: 2
attribute:	MINUTE
entity:	MINUTE (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	MONTH
entity:	MONTH (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	NAME
entity:	COURSE
properties:	Mandatory Basic Text Length: 22
attribute:	NAME_OF_SPONSOR
entity:	STUDENT
properties:	Optional Basic Text Length: 23
attribute:	NEXT_DUTY_STATION
entity:	STUDENT
properties:	Optional Basic Text Length: 20
attribute:	NEXT_FITREP_DUE
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	NPS_MAJOR
entity:	STUDENT
properties:	Optional Basic Text Length: 25

Attribute Cross Reference

attribute:	NUMBER
entity:	NUMTIME (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	NUMBER
entity:	NUMCYY (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	NUMBER
entity:	CURRICULUM
properties:	Mandatory Basic Text Length: 3
attribute:	NUMBER
entity:	COURSE
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	NUMBER
entity:	THESIS
properties:	Mandatory Designed Number Length: 6 Decimal: 0
attribute:	NUMBER
entity:	CYNUM (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	NUMBER
entity:	NUMDATE (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	NUMBER
entity:	NUMTEXT (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 15 Decimal: 0
attribute:	NUMBER
entity:	TEXTNUM (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 15 Decimal: 0
attribute:	NUMBER_ACADEMIC_QTRS_AUTHORIZED
entity:	STUDENT_BOOK_REIMBURSEMENT
properties:	Mandatory Designed Number Length: 1 Decimal: 0
attribute:	NUMBER DATE
entity:	DATENUM (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	NUMBER_OF_DAYS
entity:	DAYS (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0

Attribute Cross Reference

attribute:	NUMBER_OF_DAYS
entity:	DATEDAYS (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	NUMBER_TIME
entity:	TIMENUM (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	OFFICER_DESIGNATOR
subtype:	NAVY of entity: STUDENT
properties:	Optional Basic Number Length: 4 Decimal: 0
attribute:	OFFICER_YEAR_GROUP
subtype:	NAVY of entity: STUDENT
properties:	Optional Basic Number Length: 2 Decimal: 0
attribute:	OUTPUT_STRING
entity:	CONCAT (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	PASSWORD
entity:	CURRICULAR_OFFICE
properties:	Optional Basic Text Length: 8
attribute:	PASS_FAIL
entity:	STUDENT_COURSE_OF_STUDY
properties:	Mandatory Basic Text Length: 1
attribute:	PERCENTAGE
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 3 Decimal: 0
attribute:	PHONE_NUMBER
entity:	STUDENT
properties:	Optional Basic Number Length: 7 Decimal: 0
attribute:	PHYSICAL_DATE
entity:	STUDENT
properties:	Optional Basic Date Length: 8
attribute:	PLACE_OF_BIRTH_CITY
entity:	STUDENT
properties:	Optional Basic Text Length: 20
attribute:	PLACE_OF_BIRTH_STATE
entity:	STUDENT
properties:	Optional Designed Text Length: 2

Attribute Cross Reference

attribute:	PRESENT STATUS
entity:	STUDENT
properties:	Mandatory Basic Text Length: 10
attribute:	PREVIOUS_DUTY_STATION
entity:	STUDENT
properties:	Optional Basic Text Length: 20
attribute:	PROGRAM
subtype:	CIVILIAN of entity: STUDENT
properties:	Mandatory Designed Text Length: 17
attribute:	PROPOSED_NPS_DEGREE
entity:	STUDENT
properties:	Optional Basic Text Length: 30
attribute:	QUARTER NUMBER
entity:	QTR_OF_TYPICAL_STUDY
properties:	Mandatory Basic Number Length: 2 Decimal: 0
attribute:	RANK
entity:	STUDENT
properties:	Mandatory Basic Text Length: 2
attribute:	RECEIVED_ORDERS_TO_ATTEND
entity:	STUDENT
properties:	Mandatory Basic Text Length: 1
attribute:	REFRESHER_REQUIREMENTS
entity:	TYPICAL_COURSE_OF_STUDY
properties:	Mandatory Basic Number Length: 1 Decimal: 0
attribute:	SCHOOL
entity:	ACADEMIC_HISTORY
properties:	Optional Basic Text Length: 22
attribute:	SCORE
entity:	PRT
properties:	Mandatory Basic Text Length: 1
attribute:	SECOND
entity:	SECOND (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	SECOND_READER
entity:	THESIS
properties:	Optional Basic Text Length: 22

Attribute Cross Reference

attribute:	SECTION_NUMBER
subtype:	SCHEDULED of entity: STUDENT_COURSE_OF_STUDY
properties:	Mandatory Basic Text Length: 1
attribute:	SECTION_NUMBER
entity:	STUDENT
properties:	Optional Basic Text Length: 6
attribute:	SECURITY_ACCESS
entity:	STUDENT
properties:	Optional Basic Text Length: 1
attribute:	SECURITY_BACKGROUND
entity:	STUDENT
properties:	Optional Basic Text Length: 22
attribute:	SERVICE
subtype:	MILITARY_NON_NAVY of entity: STUDENT
properties:	Mandatory Basic Text Length: 4
attribute:	SHORTNAME
entity:	STUDENT
properties:	Optional Basic Text Length: 8
attribute:	SMC_BOX_NUMBER
entity:	STUDENT
properties:	Optional Basic Number Length: 4 Decimal: 0
attribute:	SPLIT_SECTION
entity:	STUDENT
properties:	Optional Basic Text Length: 2
attribute:	SSN
entity:	STUDENT
properties:	Mandatory Basic Text Length: 9
attribute:	STARTED_PARENT_CURRICULUM
entity:	STUDENT
properties:	Mandatory Basic Text Length: 2
attribute:	STARTING_POSITION
entity:	SUBSTR (available ief entity & attribute)
properties:	Optional Basic Number Length: 4 Decimal: 0
attribute:	STATUS
entity:	STUDENT_COURSE_OF_STUDY
properties:	Mandatory Basic Text Length: 1

Attribute Cross Reference

attribute:	STATUS
entity:	THESIS
properties:	Mandatory Basic Text Length: 1
attribute:	STREET
entity:	STUDENT
properties:	Optional Basic Text Length: 40
attribute:	STRING
entity:	NUMTEXT (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 17
attribute:	STRING
entity:	UPPER (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	STRING
entity:	TEXTNUM (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 15
attribute:	STRING
entity:	FIND (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	STRING
entity:	VERIFY (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	STRING1
entity:	CONCAT (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	STRING2
entity:	CONCAT (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	STRING_LENGTH
entity:	LENGTH (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 9 Decimal: 0
attribute:	STUDY_SPACE
entity:	STUDENT
properties:	Optional Basic Text Length: 22
attribute:	SUBSCRIPT
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 9 Decimal: 0

Attribute Cross Reference

attribute:	SUBSTRING
entity:	FIND (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	SUBSTRING
entity:	SUBSTR (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	SUBSTRING_LENGTH
entity:	SUBSTR (available ief entity & attribute)
properties:	Optional Basic Number Length: 4 Decimal: 0
attribute:	TIME_STRING
entity:	TIMETEXT (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 8
attribute:	TIME_VALUE
entity:	HOUR (available ief entity & attribute)
properties:	Mandatory Basic Time Length: 6
attribute:	TIME_VALUE
entity:	MINUTE (available ief entity & attribute)
properties:	Mandatory Basic Time Length: 6
attribute:	TIME_VALUE
entity:	SECOND (available ief entity & attribute)
properties:	Mandatory Basic Time Length: 6
attribute:	TIME_VALUE
entity:	TIMENUM (available ief entity & attribute)
properties:	Mandatory Basic Time Length: 6
attribute:	TIME_VALUE
entity:	NUMTIME (available ief entity & attribute)
properties:	Mandatory Basic Time Length: 6
attribute:	TIME_VALUE
entity:	TIMETEXT (available ief entity & attribute)
properties:	Mandatory Basic Time Length: 6
attribute:	TITLE
entity:	THESIS
properties:	Optional Basic Text Length: 254
attribute:	TITLE
entity:	CURRICULAR_OFFICE
properties:	Mandatory Basic Text Length: 65

Attribute Cross Reference

attribute:	TITLE
entity:	CURRICULUM
properties:	Mandatory Basic Text Length: 40
attribute:	TOTAL
entity:	QPR (local work element creation)
properties:	Mandatory Basic Number Length: 6 Decimal: 2
attribute:	TOTAL
entity:	QUARTER_QPR
properties:	Mandatory Derived Number Length: 4 Decimal: 2
attribute:	TOTAL_AMOUNT
entity:	TOTAL_ANNUAL NAVAL BOOK CEILING
properties:	Mandatory Basic Number Length: 6 Decimal: 2
attribute:	TOTAL_AMOUNT_ELIGIBLE
entity:	STUDENT_BOOK_REIMBURSEMENT
properties:	Mandatory Basic Number Length: 6 Decimal: 2
attribute:	TOTAL_CURRENCY
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 15 Decimal: 2
attribute:	TOTAL_INTEGER
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 15 Decimal: 0
attribute:	TOTAL_QPR
entity:	STUDENT
properties:	Optional Derived Number Length: 4 Decimal: 2
attribute:	TOTAL_REAL
entity:	IEF_SUPPLIED
properties:	Mandatory Basic Number Length: 15 Decimal: 4
attribute:	TRIMMED_STRING
entity:	TRIM (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	TYPE_OFFICER
entity:	STUDENT
properties:	Mandatory Basic Text Length: 1
attribute:	TYPE_OF_COURSE
entity:	COMPOSITION_OF_TYPICAL_STUDY
properties:	Mandatory Basic Text Length: 1

Attribute Cross Reference

attribute:	TYPE_REFRESHER
entity:	STUDENT
properties:	Mandatory Basic Number Length: 1 Decimal: 0
attribute:	TYPE_STUDENT
entity:	TYPICAL COURSE OF STUDY
properties:	Mandatory Basic Text Length: 1
attribute:	UPPER_STRING
entity:	UPPER (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	VALIDATION
entity:	STUDENT COURSE OF STUDY
properties:	Mandatory Basic Text Length: 1
attribute:	VALIDATION_STRING
entity:	VERIFY (available ief entity & attribute)
properties:	Mandatory Basic Text Length: 4096
attribute:	VALUE
entity:	POINT (local work element creation)
properties:	Mandatory Basic Number Length: 2 Decimal: 1
attribute:	VERIFY
entity:	VERIFY (available ief entity & attribute)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	YEAR
entity:	STUDENT BOOK REIMBURSEMENT
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	YEAR
entity:	YEAR (local work element creation)
properties:	Mandatory Basic Number Length: 4 Decimal: 0
attribute:	ZIP_CODE
entity:	STUDENT
properties:	Optional Basic Number Length: 9 Decimal: 0

-End of Report-

APPENDIX E

The report on the following pages presents the Attribute Definition of the designed system. This report contains information about the attributes that are specified in the data model, such as attribute name, and aliases, entity type, description, properties, length, default value, permitted values, and permitted value descriptions. Attributes are listed in the alphabetical order of their associated entity types. [Ref. 27:p. 32-9]

Attribute Definition

Entity Type: ACADEMIC_HISTORY

=====

Attribute: SCHOOL

Description: Name of undergraduate or graduate school

Properties: Optional Basic Text

Length: 22

Attribute: DEGREE

Description: Undergraduate or graduate degree earned

Properties: Mandatory Basic Text

Length: 15

Attribute: MAJOR

Description: Undergraduate or graduate major earned

Properties: Mandatory Basic Text

Length: 15

Attribute: GPA

Description: Undergraduate or graduate grade point average for the degree/major earned

Properties: Optional Basic Number

Length: 3

Decimal places: 2

Default: none

Permitted Values

0.00 to 4.00

Attribute: DATE

Description: Date undergraduate or graduate degree was awarded

Properties: Optional Basic Date

Length: 8

Attribute Definition

Entity Type: BOOK_CLAIM

Attribute: AMOUNT_OF_CLAIM

Description: Represents the amount of text books claimed by the
Naval student for a given Academic Quarter

Properties: Mandatory Basic Number

Length: 5

Decimal places: 2

Attribute: ACADEMIC_QUARTER

Description: Identifies the particular academic quarter of a
particular book claim

Properties: Mandatory Basic Text

Length: 4

Default: none

Permitted Values

FALL

WIN

SUM

SPR

Attribute Definition

Entity Type: COMPOSITION_OF_TYPICAL_STUDY

=====

Attribute: TYPE_OF_COURSE

Description: Indicates if a recommended course is an Elective or a
Required course

Properties: Mandatory Basic Text

Length: 1

Default Value: R

Permitted Values

E	Elective
R	Required

Attribute Definition

Entity Type: COURSE

=====

Attribute: NAME

Description: Indicates the actual name of the course

Properties: Mandatory Basic Text

Length: 22

Attribute: LECTURE_CREDIT_HOURS

Description: Number of hours of classroom instruction for a given course

Properties: Mandatory Basic Number

Length: 1

Decimal places: 0

Default: none

Permitted Values

0 to 5

Attribute: LAB_CREDIT_HOURS

Description: Number of laboratory hours for a given course

Properties: Mandatory Basic Number

Length: 1

Decimal places: 0

Default Value: 0

Permitted Values

0 to 5

Attribute: ACADEMIC_DEPARTMENT_CODE

Description: Indicates the two digit code for the academic department which teaches a particular course

Properties: Mandatory Basic Text

Length: 2

Default: none

Permitted Values

SE	PH
OA	GH
OS	OC
NS	MR
ME	MA
MS	EW
EC	CS
ST	AE
CC	MN
IS	CO
CM	AS

Entity Type: COURSE

Attribute: NUMBER

Description: Indicates the four digit number identifying the
graduate level of a particular course

Properties: Mandatory Basic Number

Length: 4

Decimal places: 0

Default: none

Permitted Values

0001 to 4999

Attribute Definition

Entity Type: CURRICULAR_OFFICE

=====

Attribute: TITLE

Description: The title of one of the 11 curricular offices

Properties: Mandatory Basic Text

Length: 65

Attribute: CODE

Description: The two digit code used to identify a NPS curricular office

Properties: Mandatory Basic Text

Length: 2

Default: none

Permitted Values

3A	39
38	37
36	35
34	33
32	31
30	

Attribute: PASSWORD

Description: The password used by the Management of Information System's office to restrict access to a particular curricular officer's student records.

Properties: Optional Basic Text

Length: 8

Attribute Definition

Entity Type: CURRICULUM

=====

Attribute: TITLE

Description: The descriptive title of a particular curriculum

Properties: Mandatory Basic Text

Length: 40

Attribute: NUMBER

Description: Three digit number identifying a particular curriculum

Properties: Mandatory Basic Text

Length: 3

Attribute Definition

Entity Type: DEPENDENT

Attribute: LAST_NAME

Description: Indicates the last name of a student's dependent

Properties: Mandatory Basic Text

Length: 23

Attribute: FIRST_NAME

Description: Indicates the first name of a student's dependent

Properties: Mandatory Basic Text

Length: 15

Attribute: FAMILY_MEMBER

Description: Identifies the family relationship of a dependent (ie Spouse, Child)

Properties: Mandatory Basic Text

Length: 6

Default: none

Permitted Values

SPOUSE

CHILD

Attribute: ALSO_STUDENT

Subtype: SPOUSE

Description: Indicates if the spouse is also enrolled as a student at the Naval Postgraduate School

Properties: Optional Basic Text

Length: 1

Default Value: N

Permitted Values

N

Y

Attribute: DATE_OF_BIRTH

Subtype: CHILD

Description: Indicates the date of birth of a child

Properties: Optional Basic Date

Length: 8

Entity Type: DEPENDENT

Attribute: GENDER

Subtype: CHILD

Description: Indicates the sex of the child

Properties: Optional Designed Text

Length: 1

Default: none

Permitted Values

F

M

Attribute Definition

Entity Type: PRT

=====

Attribute: SCORE

Description: Indicates the letter score of a particular physical readiness test (ie. Outstanding, Excellent, Good, Satisfactory, or Failure)

Properties: Mandatory Basic Text

Length: 1

Default: none

Permitted Values

F	FAIL!
S	Satisfactory
G	Good
E	Excellent
O	Outstanding

Attribute: BODY_FAT

Description: Indicates the body fat percentage

Properties: Optional Basic Number

Length: 2

Decimal places: 0

Attribute: DATE_OF_TEST

Description: Date the physical readiness test was taken

Properties: Mandatory Basic Date

Length: 8

Attribute Definition

Entity Type: QTR_OF_TYPICAL_STUDY

=====

Attribute: QUARTER_NUMBER

Description: Identifies the numeric quarter of a recommended course

Properties: Mandatory Basic Number

Length: 2

Decimal places: 0

Attribute Definition

Entity Type: QUARTER_QPR

=====

Attribute: TOTAL

Description: When the quarter-hour credit of a particular quarter's course is multiplied by the point value of the student's grade, a quality point value for the student's work in the course for that quarter is obtained. The sum of quality points for all courses taken that quarter are divided by the sum of quarter-hour credits of these courses gives a weighted numerical evaluation of the student's performance.

Properties: Mandatory Derived Number

Length: 4

Decimal places: 2

Default Algorithm: CALCULATE_QTR_TOTAL_QPR

Attribute: GRADUATE

Description: When the quarter-hour credit of a particular quarter's graduate course is multiplied by the point value of the student's grade, a quality point value for the student's work in the graduate course for that quarter is obtained. The sum of the quality points for all graduate courses divided by the sum of the quarter-hour credit of these courses gives a weighted numerical evaluation of the student's performance.

Properties: Mandatory Derived Number

Length: 4

Decimal places: 2

Default Algorithm: CALCULATE_QTR_GRADUATE_QPR

Attribute: ACADEMIC_YEAR

Description: Academic year (fiscal year) of a particular quarter

Properties: Mandatory Basic Number

Length: 2

Decimal places: 0

Attribute: ACADEMIC_QUARTER

Description: The academic quarter (season) of a particular year (ie. Fall, Winter, Spring, Summer)

Properties: Mandatory Basic Text

Length: 4

Default: none

Permitted Values

FALL
SUM
SPR
WIN

Attribute Definition

Entity Type: STUDENT

=====

Attribute: SSN

Description: The student's social security number; International students have an alphanumeric code (ex. TKN9126785)

Properties: Mandatory Basic Text

Length: 9

Attribute: TYPE_REFRESHER

Description: The type of refresher a student will require. (1 for direct input: no refresher; 2 for a direct input but 6 week refresher requirement; 3 for a 460 curriculum enrollment with one quarter of refresher required; and 4 for a 460 curriculum enrollment of two quarter refresher required.)

Properties: Mandatory Basic Number

Length: 1

Decimal places: 0

Default Value: 2

Permitted Values

- | | |
|---|---|
| 4 | 4 - requires 2 quarter refresher under Engineering Science curriculum (460) |
| 3 | 3 - requires 1 quarter refresher under Engineering Science Curriculum (460) |
| 2 | 2 - direct input (requires 6 week refresher course) |
| 1 | 1 - direct input (does not require a refresher) |

Attribute: PRESENT_STATUS

Description: This represents the lifecycle of a student in the following order of occurrence: Projected; Arrived; Registered; Graduated or Dropped

Properties: Mandatory Basic Text

Length: 10

Default Value: A

Permitted Values

- | | | |
|---|-------------|--|
| A | Arrived: | student has arrived, but has not been officially registered by the registrar |
| P | Projected: | future-student; expected arrival |
| R | Registered: | student officially registered by the registrar |
| G | Graduated: | can only be entered on or after the date of graduation |
| D | Dropped: | student did not complete required courses for a degree |

Entity Type: STUDENT

Attribute: TYPE OFFICER

Description: Describes the type of officer as: International,
Civilian, Naval, or Military (ie. Non-Naval)

Properties: Mandatory Basic Text

Length: 1

Default Value: N

Permitted Values

M Military (non-Navy)
N Navy
C Civilian
I International

Attribute: RECEIVED_ORDERS_TO_ATTEND

Description: This attribute indicates if a notice of acceptance or
orders directing a prospective student has been
received.

Properties: Mandatory Basic Text

Length: 1

Default: none

Permitted Values

N
Y

Attribute: LAST_NAME

Description: Student's last name

Properties: Mandatory Basic Text

Length: 23

Attribute: FIRST_NAME

Description: Student's first name

Properties: Mandatory Basic Text

Length: 15

Attribute: MIDDLE_INITIAL

Description: Student's middle initial

Properties: Optional Basic Text

Length: 2

Attribute: SHORTNAME

Description: Short version of student's name used to retrieve a
student record. (Usually consists of the first 8
characters of a student's last name)

Properties: Optional Basic Text

Length: 8

Entity Type: STUDENT

Attribute: GENDER

Description: Student's sex

Properties: Optional Basic Text

Length: 1

Default: none

Permitted Values

M

F

Attribute: RANK

Description: Military pay grade of a student. (O1 to O6)

Properties: Mandatory Basic Text

Length: 2

Default: none

Permitted Values

O1 ENS (Navy) or 2NDLT
O2 LTJG (Navy) or 1STLT
O3 LT (Navy) or CAPT
O4 LCDR (Navy) or MAJ
O5 CDR (Navy) or LTCOL
O6 CDR (Navy) or COL

Attribute: DATE_OF_RANK

Description: Student's date of current military rank

Properties: Optional Basic Date

Length: 8

Attribute: PHONE_NUMBER

Description: Student's home phone number

Properties: Optional Basic Number

Length: 7

Decimal places: 0

Attribute: STREET

Description: Street address of student's current residence

Properties: Optional Basic Text

Length: 40

Attribute: CITY

Description: City of student's current residence

Properties: Optional Basic Text

Length: 22

Attribute: ZIP_CODE

Description: Zip code of student's current residence

Properties: Optional Basic Number

Length: 9

Decimal places: 0

Entity Type: STUDENT

Attribute: LAMESA HOUSING OCCUPANT

Description: Indicates whether a student resides in Navy housing in La Mesa Village (Y/N)

Properties: Optional Basic Text

Length: 1

Default: none

Permitted Values

Y

N

Attribute: SMC_BOX_NUMBER

Description: Student's four digit student mail center box number

Properties: Optional Basic Number

Length: 4

Decimal places: 0

Attribute: DATE_REPORTED_ABOARD

Description: Date student checked into Curriculum Office

Properties: Optional Basic Date

Length: 8

Attribute: CONVENING_DATE

Description: This is the date that a student will actual start training.

Properties: Mandatory Basic Date

Length: 8

Attribute: ANTICIPATED_GRADUATION_DATE

Description: Student's prospective graduation date

Properties: Optional Basic Date

Length: 8

Attribute: SECTION_NUMBER

Description: Student's assigned curriculum section number (6 digit code)

Properties: Optional Basic Text

Length: 6

Attribute: SPLIT_SECTION

Description: Used to split the section up when it exceeds 60 students

Properties: Optional Basic Text

Length: 2

Entity Type: STUDENT

Attribute: PROPOSED_NPS_DEGREE

Description: Degree a student will receive upon graduation; see Accreditation Status for students listed in Engineer.

Properties: Optional Basic Text

Length: 30

Default Value: MASTER OF SCIENCE

Permitted Values

DOCTORATE

ENGINEER

MASTER OF SCIENCE

MASTER OF ARTS

Attribute: ACCREDITATION_STATUS

Description: This attribute indicates the status of Accreditation for a student in an Engineer Degree.

Properties: Optional Basic Text

Length: 1

Default: none

Permitted Values

C Complete

I In progress

N Not evaluated

Attribute: NPS_MAJOR

Description: Student's academic major at NPS (Major Specialty)

Properties: Optional Basic Text

Length: 25

Attribute: APC

Description: Three digit code indicating a student's academic profile code

Properties: Optional Basic Number

Length: 3

Decimal places: 0

Attribute: STUDY_SPACE

Description: Used to note areas where the student studies when not in class

Properties: Optional Basic Text

Length: 22

Attribute: LOCKER_NUMBER

Description: Indicates the student's locker number

Properties: Optional Basic Number

Length: 3

Decimal places: 0

Entity Type: STUDENT

Attribute: MARITAL STATUS

Description: Indicates if a student is married, single, or divorced

Properties: Optional Basic Text

Length: 1

Default: none

Permitted Values

M	married
S	single
D	divorced

Attribute: DATE OF BIRTH

Description: Date student was born

Properties: Optional Basic Date

Length: 8

Attribute: PLACE OF BIRTH CITY

Description: City student was born

Properties: Optional Basic Text

Length: 20

Attribute: PLACE OF BIRTH STATE

Description: State student was born

Properties: Optional Designed Text

Length: 2

Attribute: SECURITY BACKGROUND

Description: Last security clearance and who granted the clearance
(ie. NAC/901214)

Properties: Optional Basic Text

Length: 22

Attribute: SECURITY ACCESS

Description: Single character that indicates No clearance,
Confidential, Secret or Top Secret clearance level

Properties: Optional Basic Text

Length: 1

Default: none

Permitted Values

N	No Clearance
C	Confidential
S	Secret
T	Top Secret

Attribute: COMMISSIONING SOURCE

Description: Student's commissioning source (ie. NROTC, USNA, etc)

Properties: Optional Basic Text

Length: 6

Entity Type: STUDENT

Attribute: DUAL_DEGREE

Description: Indicates the name of the second degree. Blank indicates no dual degree.

Properties: Optional Basic Text

Length: 25

Attribute: LAST_FITREP_DATE

Description: Ending date of last fitness report. (Needed to ensure continuity of the fitness report)

Properties: Optional Basic Date

Length: 8

Attribute: NEXT_FITREP_DUE

Description: Beginning date of when the next fitness report is due

Properties: Optional Basic Date

Length: 8

Attribute: PREVIOUS_DUTY_STATION

Description: Name of previous duty station

Properties: Optional Basic Text

Length: 20

Attribute: NEXT_DUTY_STATION

Description: Name of next duty station

Properties: Optional Basic Text

Length: 20

Attribute: DATE_OF_ORDERS

Description: Date of orders stated on orders message

Properties: Optional Basic Date

Length: 8

Attribute: NAME_OF_SPONSOR

Description: Last name of sponsor assigned to a student

Properties: Optional Basic Text

Length: 23

Attribute: DATE_THAT_A_SPONSOR_WAS_ASSIGNED

Description: Date sponsor was assigned to a student

Properties: Optional Basic Date

Length: 8

Attribute: DATE_WELCOME_PACKAGE_SENT

Description: Date Welcome Aboard Package was sent to prospective student

Properties: Optional Basic Date

Length: 8

Entity Type: STUDENT

Attribute: DATE SPONSOR LETTER SENT

Description: Date Sponsor Letter was sent to prospective student

Properties: Optional Basic Date

Length: 8

Attribute: LIBRARY_CARD_NUMBER

Description: Student's Library Card number

Properties: Optional Basic Number

Length: 7

Decimal places: 0

Attribute: MAINFRAME_ACCOUNT_NUMBER

Description: Student's Mainframe Account Number

Properties: Optional Basic Text

Length: 5

Attribute: AIDS_TEST_DATE

Description: Date of last AIDS test

Properties: Optional Basic Date

Length: 8

Attribute: PHYSICAL_DATE

Description: Date of last physical exam

Properties: Optional Basic Date

Length: 8

Attribute: DENTAL_DATE

Description: Date of last dental exam

Properties: Optional Basic Date

Length: 8

Attribute: IN_BOUND_STUDENT_SPONSOR

Description: Indicates if the student has been assigned as a sponsor for an inbound student

Properties: Optional Basic Text

Length: 1

Default Value: N

Permitted Values

Y

N

Entity Type: STUDENT

Attribute: TOTAL_QPR

Description: Total Quality Point Rating: When the quarter-hour credit of a course is multiplied by the point value of the student's grade, a quality point value for the student's work in the course is obtained. The sum of the quality points for all courses divided by the sum of the quarter-hour credit of these courses gives a weighted numerical evaluation.

Properties: Optional Derived Number

Length: 4

Decimal places: 2

Default Algorithm: CALCULATE_TOTAL_QPR

Attribute: GRADUATE_QPR

Description: Graduate Courses Quality Point Rating: When the quarter-hour credit of a graduate course is multiplied by the point value of the student's grade, a quality point value for the student's work in the course is obtained. The sum of the quality points for all graduate courses divided by the sum of the quarter-hour credit of these courses gives a weighted numerical evaluation of a student's performance.

Properties: Optional Derived Number

Length: 4

Decimal places: 2

Default Algorithm: CALCULATE_GRADUATE_QPR

Attribute: STARTED_PARENT_CURRICULUM

Description: This attribute indicates whether a student has completed the 460 curriculum; transparent to user as this attribute is used and derived by the system only.

Properties: Mandatory Basic Text

Length: 2

Default Value: NA

Permitted Values

NA	This indicates that a student is NOT enrolled in 460 curriculum (ie - therefore, student is a direct input); system is not concerned with 6 week refresher in order to account for students enrolled in 460 curriculum.
N	This indicates that a student is presently enrolled in 460 curriculum and therefore, has NOT yet started regular curriculum; student's Type_Refresher may be either 1 Qtr or 2 Qtrs (type 3 or 4, respectively).
Y	This indicates that a student has completed the 460 curriculum and is presently enrolled in their parent curriculum.

Entity Type: STUDENT

Attribute: COMPLETED_FIRST_REFRESHER_QTR

Description: This attribute is used to determine whether a student has completed the first quarter of a two quarter (ie. Type_Refresher = 4) 460 curriculum; transparent to user as this attribute is used and derived by the system only.

Properties: Mandatory Basic Text

Length: 2

Default Value: NA

Permitted Values

NA	This value indicates that a student is NOT enrolled in the two quarter 460 curriculum (ie - therefore, student's Type_Refresher = 3 as a one quarter 460 curriculum or is = Type 1 or 2 as a direct input);
N	This value indicates that a student is presently enrolled in 460 curriculum and therefore, has NOT yet completed the first quarter of a two quarter refresher; student's Type_Refresher is = 4 only.
Y	This value indicates that a student has completed the first quarter of a two quarter 460 curriculum.

Attribute: COMMENT1

Description: Thirty character remarks section

Properties: Optional Basic Text

Length: 30

Attribute: PROGRAM

Subtype: CIVILIAN

Description: Indicates the type of program a civilian will participate in

Properties: Mandatory Designed Text

Length: 17

Default Value: REGULAR CURRICULA

Permitted Values

REGULAR CURRICULA
DEGREE PROGRAM
NON DEGREE PGM

Attribute: INTERNATIONAL_SERVICE_COMPONENT

Subtype: INTERNATIONAL

Description: Indicates the Military Service Component of an International Officer

Properties: Mandatory Basic Text

Length: 5

Entity Type: STUDENT

Attribute: COUNTRY

Subtype: INTERNATIONAL

Description: Identifies the country of an International student

Properties: Optional Designed Text

Length: 10

Attribute: SERVICE

Subtype: MILITARY_NON_NAVY

Description: Identifies the Military Service of a Non-Naval Student

Properties: Mandatory Basic Text

Length: 4

Attribute: OFFICER_DESIGNATOR

Subtype: NAVY

Description: This is a four digit number indicating a US Navy Student's designator (ie. 1110, 1115)

Properties: Optional Basic Number

Length: 4

Decimal places: 0

Attribute: LINEAL_NUMBER

Subtype: NAVY

Description: US Navy student's eight character lineal number

Properties: Optional Basic Number

Length: 8

Decimal places: 0

Attribute: OFFICER_YEAR_GROUP

Subtype: NAVY

Description: Student's two digit year group number

Properties: Optional Basic Number

Length: 2

Decimal places: 0

Attribute Definition

Entity Type: STUDENT_BOOK_REIMBURSEMENT

=====

Attribute: AMOUNT_REMAINING

Description: Maintained for each Navy student record in order to enhance response time of system for viewing and reporting; performance factors over storage considerations; calculate without increasing wait time

Properties: Mandatory Derived Number

Length: 6

Decimal places: 2

Default Algorithm: AMOUNT_REMAINING

Attribute: TOTAL_AMOUNT_ELIGIBLE

Description: Indicates the total amount a student is authorized to use for a particular year (this attribute is derived)

Properties: Mandatory Basic Number

Length: 6

Decimal places: 2

Attribute: NUMBER_ACADEMIC_QTRS_AUTHORIZED

Description: Indicates the number of quarters authorized for a particular year

Properties: Mandatory Designed Number

Length: 1

Decimal places: 0

Default Algorithm: DETERMINE_NUMBER_ACADEMIC_QTRS

Attribute: YEAR

Description: Indicates the academic year of a particular student's book reimbursement

Properties: Mandatory Basic Number

Length: 4

Decimal places: 0

Attribute Definition

Entity Type: STUDENT COURSE OF STUDY

=====

Attribute: STATUS

Description: Indicates if a particular course has been requested,
scheduled, or completed.

Properties: Mandatory Basic Text

Length: 1

Default Value: R

Permitted Values

C	Completed
S	Scheduled
R	Requested

Attribute: VALIDATION

Description: Indicates if a course has been validated

Properties: Mandatory Basic Text

Length: 1

Default Value: N

Permitted Values

Y
N

Attribute: PASS FAIL

Description: Indicates if a course is to be taken as for a pass/
fail grade

Properties: Mandatory Basic Text

Length: 1

Default Value: N

Permitted Values

Y
N

Attribute: ACADEMIC YEAR

Description: The academic year (fiscal year) of a particular
quarter

Properties: Mandatory Basic Number

Length: 4

Decimal places: 0

Entity Type: STUDENT COURSE OF STUDY

Attribute: ACADEMIC_QUARTER

Description: The academic quarter (season) of a particular year
(ie. Fall, Winter, Spring, Summer)

Properties: Mandatory Basic Text

Length: 4

Default: none

Permitted Values

WIN
SPR
SUM
FALL

Attribute: GRADE

Subtype: COMPLETED

Description: Indicates the letter grade earned for a particular
course

Properties: Mandatory Designed Text

Length: 2

Default Value: I

Permitted Values

F Fail
P Pass
N Ungraded
W Withdrew
I Incomplete
X If an "I" is not removed within the twelve weeks
following the end of the quarter it becomes an
"X"; where X = 0.00 point value
D
D+
C-
C
C+
B-
B
B+
A-
A

Attribute: SECTION NUMBER

Subtype: SCHEDULED

Description: Indicates the section number of a scheduled course

Properties: Mandatory Basic Text

Length: 1

Attribute Definition

Entity Type: THESIS

=====

Attribute: NUMBER

Description: This number is system generated to identify the Thesis of a specific year. Once a thesis proposal is submitted, this entity will be created.

Properties: Mandatory Designed Number

Length: 6

Decimal places: 0

Default Algorithm: THESIS_NUMBER

Attribute: DUE_DATE_MONTH

Description: This identifies the month that a thesis is to be submitted. An extension which would change the year of the recorded Thesis, must be canceled entirely and re-entered with a new thesis number.

Properties: Mandatory Basic Text

Length: 3

Default: none

Permitted Values

DEC	NOV
OCT	SEP
AUG	JUL
JUN	MAY
APR	MAR
FEB	JAN

Attribute: DUE_DATE_YEAR

Description: This identifies the year that a thesis will be submitted. Should the year change, the thesis would have to be deleted and renumbered, since numbers are assigned on a yearly basis.

Properties: Mandatory Basic Number

Length: 2

Decimal places: 0

Attribute: TITLE

Description: Describes the student's particular thesis

Properties: Optional Basic Text

Length: 254

Entity Type: THESIS

Attribute: STATUS

Description: Indicates whether student's thesis is completed yet
(ie. Yes, No, or extension)

Properties: Mandatory Basic Text

Length: 1

Default Value: N

Permitted Values

E extension
N not completed
Y has been completed

Attribute: ADVISOR

Description: Name of the Thesis Advisor

Properties: Optional Basic Text

Length: 22

Attribute: SECOND READER

Description: Name of the second reader

Properties: Optional Basic Text

Length: 22

Attribute: CLASSIFIED

Description: Indicates if a particular thesis is Classified or
Unclassified

Properties: Mandatory Basic Text

Length: 1

Default Value: U

Permitted Values

C Classified
U Unclassified

Attribute: JOINT

Description: Indicates whether a thesis is being jointly written

Properties: Mandatory Basic Text

Length: 1

Default Value: N

Permitted Values

N
Y

Attribute Definition

Entity Type: TOTAL_ANNUAL_NAVAL_BOOK_CEILING

=====

Attribute: TOTAL_AMOUNT

Description: This is the total amount authorized annual for the reimbursement for books purchased by Naval Students.

Properties: Mandatory Basic Number

Length: 6

Decimal places: 2

Attribute: DATE_IMPLEMENTED

Description: This is the effective date of the Naval Regulation book reimbursement ceiling.

Properties: Mandatory Basic Date

Length: 8

Attribute Definition

Entity Type: TYPICAL_COURSE_OF_STUDY

Attribute: TYPE_STUDENT

Description: Indicates the type of officer (Naval, International, Civilian, or Military (non-naval)) a particular course of study is recommended

Properties: Mandatory Basic Text

Length: 1

Default Value: N

Permitted Values

C	Civilian
M	Military (non-Navy)
N	Navy
I	International

Attribute: REFRESHER_REQUIREMENTS

Description: Indicates the type of refresher requirements a particular course of study is recommended

Properties: Mandatory Basic Number

Length: 1

Decimal places: 0

Default: none

Permitted Values

4	4 is two quarters of Engineering Science (460)
3	3 is one quarter of Engineering Science (460)
2	2 is technical refresher (6 week)
1	1 is direct input (ie - no refresher courses required)

-End of Report-

APPENDIX F

The report on the following pages presents the Activity Definition (or as referenced in earlier versions of IEF, Process Definition) of the designed system. This report contains a description of the functions and processes of the activity model. The expected effects such as creation, update or deletion of elementary processes are defined in this report. [Ref. 27:p. 32-10]

Activity Definition

Name: ACADEMIC_COUNSELING

Description: This function involves the maintenance of the academic records of a student in a particular curricular office.

Type: Function

Subordinate of: SUPERVISE_ENROLLED_STUDENT

Subordinates: RECORD_ARRIVAL
SETUP_STUDENT_COURSE_OF_STUDY
CHANGE_COURSE_IN_STUDENT_STUDY
CHANGE_REQT_OF_COURSE_REQUEST
RECORD_THESIS_PROPOSAL
REVISE_THESIS_PROPOSAL
ELIMINATE_JOINT_STATUS
REVISE_THESIS_TO_JOINT_STATUS
ARCHIVE_THESIS
REMOVE_ERRONEOUS_THESIS

Activity Definition

Name: ACADEMIC_DATABASE_ADMINISTRATION

Description: This function incorporates the interaction of academic student database.

Type: Function

Subordinates: INITIALIZE_STUDENT_RECORD
COUNSELING_FUTURE_STUDENT
SUPERVISE_ENROLLED_STUDENT
COMPLETED_ACADEMIC_REQUIREMENTS
CURRICULUM_DEVELOPMENT_MGMT
COURSE_MAINTENANCE
SYSTEM_MANAGEMENT

Activity Definition

Name: ARCHIVE_THESIS

Description: System Gen: This process archives thesis listing for those students who have been archived. (Outside scope: Thesis Processor)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type

THESIS

Expected Actions

read
delete

Process ARCHIVE_THESIS

Import Views

View INPUT of entity THESIS
Attributes:

DUE_DATE_YEAR
NUMBER

Export Views

View OUTPUT_REMOVED of entity THESIS
Attributes:

NUMBER
DUE_DATE_YEAR

Entity Action Views

View of entity THESIS
Attributes:

NUMBER
DUE_DATE_YEAR

Activity Definition

Name: ASSIGN_NAVY_BOOK_CEILING

Description: System Gen: This process creates the Naval ceiling for the Naval Book Eligibility. (Outside scope: MIS)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: SYSTEM_MANAGEMENT

Expected Effects:

Entity Type	Expected Actions
-----	-----
TOTAL_ANNUAL_NAVAL_BOOK_CEILING	create

Process ASSIGN_NAVY_BOOK_CEILING

Import Views

View INPUT of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING

Attributes:

TOTAL_AMOUNT
DATE_IMPLEMENTED

Export Views

View OUTPUT of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING

Attributes:

TOTAL_AMOUNT
DATE_IMPLEMENTED

Entity Action Views

View of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING

Attributes:

TOTAL_AMOUNT
DATE_IMPLEMENTED

Activity Definition

Name: ASSIGN_PROJECTED_STUDENT

Description: This process involves entering a prospective student into the Naval Postgraduate Schools files.
(Primarily Admission's responsibility, however, Curricular Officers may add a student)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: INITIALIZE_STUDENT_RECORD

Expected Effects:

Entity Type	Expected Actions
CURRICULUM	update
	read
STUDENT	create
	update

Process ASSIGN_PROJECTED_STUDENT

Import Views

View INPUT of entity STUDENT

Attributes:

SSN	RANK
TYPE_REFRESHER	ANTICIPATED_GRADUATION_DATE
TYPE_OFFICER	PROPOSED_NPS_DEGREE
RECEIVED_ORDERS_TO_ATTEND	NPS_MAJOR
LAST_NAME	APC
FIRST_NAME	CONVENING_DATE
opt MIDDLE_INITIAL	SHORTNAME
opt COMMENT1	opt DUAL_DEGREE
opt COMMISSIONING_SOURCE	opt PROGRAM
opt SERVICE	opt COUNTRY
opt INTERNATIONAL_SERVICE_COMPONENT	

View INPUT_RESPONSIBLE of entity CURRICULUM

Attributes:

NUMBER

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN	RANK
TYPE_REFRESHER	ANTICIPATED_GRADUATION_DATE
TYPE_OFFICER	PROPOSED_NPS_DEGREE
RECEIVED_ORDERS_TO_ATTEND	NPS_MAJOR
LAST_NAME	APC
FIRST_NAME	CONVENING_DATE
MIDDLE_INITIAL	SHORTNAME
COMMENT1	DUAL_DEGREE
COMMISSIONING_SOURCE	PROGRAM
SERVICE	COUNTRY
INTERNATIONAL_SERVICE_COMPONENT	

View OUTPUT of entity CURRICULUM

Attributes:

NUMBER

Entity Action Views

View of entity STUDENT

Attributes:

SSN	TYPE_REFRESHER
PRESENT_STATUS	TYPE_OFFICER
RECEIVED_ORDERS_TO_ATTEND	LAST_NAME
FIRST_NAME	MIDDLE_INITIAL
SHORTNAME	APC
ANTICIPATED_GRADUATION_DATE	RANK
PROPOSED_NPS_DEGREE	NPS_MAJOR
COMMENT1	
COMMISSIONING_SOURCE	
DUAL_DEGREE	
PROGRAM	
INTERNATIONAL_SERVICE_COMPONENT	
COUNTRY	
SERVICE	
CONVENING_DATE	
STARTED_PARENT_CURRICULUM	
COMPLETED_FIRST_REFRESHER_QTR	

View of entity CURRICULUM

Attributes:

NUMBER

Activity Definition

Name: ASSIGN_STUDENT_SPONSOR

Description: Each incoming student may or may not be assigned a student sponsor who will assist them. Both the incoming students records will reflect who will act as the sponsor and the sponsor's files will reflect that he/she acted as a sponsor.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COUNSELING_FUTURE_STUDENT

Expected Effects:

Entity Type

STUDENT

Expected Actions

update

read

Process ASSIGN_STUDENT_SPONSOR

Import Views

View ASSIGNED of entity STUDENT

Attributes:

SSN

View INCOMING of entity STUDENT

Attributes:

SSN

Export Views

View OUTPUT_ASSIGNED of entity STUDENT

Attributes:

SSN

LAST_NAME

FIRST_NAME

MIDDLE_INITIAL

IN_BOUND_STUDENT_SPONSOR

View OUTPUT_INCOMING of entity STUDENT

Attributes:

SSN

LAST_NAME

FIRST_NAME

MIDDLE_INITIAL

NAME_OF_SPONSOR

DATE_THAT_A_SPONSOR_WAS_ASSIGNED

Entity Action Views

View EXISTING_ASSIGNED of entity STUDENT

Attributes:

SSN
IN_BOUND_STUDENT_SPONSOR
LAST_NAME

View of entity STUDENT

Attributes:

SSN
LAST_NAME
FIRST_NAME
MIDDLE_INITIAL
NAME_OF_SPONSOR
DATE_THAT_A_SPONSOR_WAS_ASSIGNED

Activity Definition

Name: CHANGE_COURSE_IN_STUDENT_STUDY

Description: This process modifies the requested courses of a student.

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type

Expected Actions

STUDENT_COURSE_OF_STUDY

update

read

COURSE

update

read

Process CHANGE_COURSE_IN_STUDENT_STUDY

Import Views

View NEW_INPUT of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View NEW_INPUT of entity STUDENT_COURSE_OF_STUDY

Attributes:

VALIDATION

PASS_FAIL

ACADEMIC_YEAR

ACADEMIC_QUARTER

View PRESENT_INPUT of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View INPUT of entity STUDENT

Attributes:

SSN

Export Views

View OUTPUT_NEW of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

View OUTPUT_NEW of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS
VALIDATION
PASS_FAIL
ACADEMIC_YEAR
ACADEMIC_QUARTER

Entity Action Views

View NEW of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

View PRESENT of entity COURSE

Attributes:

ACADEMIC_DEPARTMENT_CODE
NUMBER

View of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

View of entity STUDENT

Attributes:

SSN

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS
VALIDATION
PASS_FAIL
ACADEMIC_YEAR
ACADEMIC_QUARTER

Activity Definition

Name: CHANGE_REQT_OF_COURSE_REQUEST

Description: This process will not request a different course, but will allow a change in requirements such as pass-fail, validation, academic quarter and year.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:	Expected Actions
Entity Type	
-----	-----
STUDENT_COURSE_OF_STUDY	update read

Process CHANGE_REQT_OF_COURSE_REQUEST

Import Views

View INPUT_NEW of entity STUDENT_COURSE_OF_STUDY
Attributes:
 VALIDATION
 PASS_FAIL
 ACADEMIC_YEAR
 ACADEMIC_QUARTER

View PRESENT_INPUT of entity COURSE
Attributes:
 NUMBER
 ACADEMIC_DEPARTMENT_CODE

View DESIGNATED of entity STUDENT
Attributes:
 SSN

Export Views

View OUTPUT_NEW of entity STUDENT_COURSE_OF_STUDY
Attributes:
 VALIDATION
 PASS_FAIL
 ACADEMIC_YEAR
 ACADEMIC_QUARTER
 STATUS

Entity Action Views

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity STUDENT

Attributes:

SSN

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

VALIDATION

PASS_FAIL

ACADEMIC_YEAR

ACADEMIC_QUARTER

STATUS

Activity Definition

Name: COUNSELING_FUTURE_STUDENT

Description: This function involves the welcoming of a future student who has been identified as possessing orders to attend the Naval Postgraduate School.

Type: Function

Subordinate of: ACADEMIC_DATABASE_ADMINISTRATION

Subordinates: ASSIGN_STUDENT_SPONSOR
SEND_SPONSOR_LETTER
SEND_WELCOME_ABOARD_PACKAGE

Activity Definition

Name: COURSE_MAINTENANCE

Description: This function incorporates creation, update and deletion of a course and offered courses. This function is executed within the Registrar's purview, however, modeling is required here to enable CSADS to present the view necessary to perform the Course related functions that fall within the Curricular Officers' area of responsibility.

Type: Function

Subordinate of: ACADEMIC_DATABASE_ADMINISTRATION

Subordinates: ESTABLISH_A_NEW_COURSE
MODIFY_EXISTING_COURSE
REMOVE_COURSE_FROM_CATALOG

Activity Definition

Name: COURSE_OF_STUDY_MAINTENANCE

Description: This function incorporates the creation, up-date, and deletion of a curriculum's Typical_Course_of_Study.

Type: Function

Subordinate of: CURRICULUM_DEVELOPMENT_MGMT

Subordinates: SETUP_TYPICAL_COURSE_OF_STUDY
MODIFY_TYPICAL_COURSE_OF_STUDY
REMOVE_TYPICAL_COURSE_OF_STUDY

Activity Definition

Name: CURRICULUM_DEVELOPMENT_MGMT

Description: This function involves the maintenance of the curriculum programs.

Type: Function

Subordinate of: ACADEMIC_DATABASE_ADMINISTRATION

Subordinates: CURRICULUM_OFFICE_MAINTENANCE
COURSE_OF_STUDY_MAINTENANCE

Activity Definition

Name: CURRICULUM_OFFICE_MAINTENANCE

Description: This function incorporates the creation, up-date, and deletion of curricular offices and curricula. This function is executed outside of the curricular office, but must be modeled here to enable use of these entity types within the CSADS.

Type: Function

Subordinate of: CURRICULUM_DEVELOPMENT_MGMT

Subordinates: ESTABLISH_NEW_CURRICULAR_OFFICE
MODIFY_CURRICULAR_OFFICE
REMOVE_CURRICULAR_OFFICE
SETUP_NEW_CURRICULUM
MODIFY_CURRICULUM
ELIMINATE_CURRICULUM

Activity Definition

Name: ELIMINATE_CURRICULUM

Description: System Gen: This process involves the removal of a curriculum from a particular curricular office.
(Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: CURRICULUM_OFFICE_MAINTENANCE

Expected Effects:

Entity Type

CURRICULUM

Expected Actions

read
delete

Process ELIMINATE_CURRICULUM

Import Views

View INPUT of entity CURRICULUM
Attributes:
NUMBER

Export Views

View OUTPUT of entity CURRICULUM
Attributes:
NUMBER

Entity Action Views

View of entity CURRICULUM
Attributes:
NUMBER

Activity Definition

Name: ELIMINATE_DEPENDENT_DATA

Description: This process involves the removal of an entity which is no longer a dependent of a student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Expected Effects:

Entity Type

Expected Actions

STUDENT
DEPENDENT

read
read
delete

Process ELIMINATE_DEPENDENT_DATA

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT of entity DEPENDENT

Attributes:

LAST_NAME

FIRST_NAME

Export Views

View OUTPUT_REMOVED of entity DEPENDENT

Attributes:

LAST_NAME

FIRST_NAME

View OUTPUT of entity STUDENT

Attributes:

SSN

Entity Action Views

View EXISTING of entity STUDENT

Attributes:

SSN

View of entity DEPENDENT

Attributes:

LAST_NAME

FIRST_NAME

Activity Definition

Name: ELIMINATE_JOINT_STATUS

Description: System Gen: This process involves the disassociation of one student from a joint thesis. (Outside scope: Thesis Processor)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type	Expected Actions
-----	-----
STUDENT	update
	read
THESIS	update
	read

Process ELIMINATE_JOINT_STATUS

Import Views

View INPUT_REMOVING of entity STUDENT
Attributes: SSN

View INPUT of entity THESIS
Attributes: DUE_DATE_YEAR
NUMBER

Export Views

View OUTPUT_REMOVED of entity STUDENT
Attributes: SSN

View OUTPUT of entity THESIS
Attributes: NUMBER
DUE_DATE_YEAR JOINT

Entity Action Views

View EXISTING of entity STUDENT
Attributes: SSN

View of entity THESIS
Attributes: NUMBER
DUE_DATE_YEAR JOINT

Activity Definition

Name: ENTER_ACADEMIC_BACKGROUND

Description: This process involves the recording of a student's prior academic history.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Expected Effects:

Entity Type

Expected Actions

STUDENT

update

read

ACADEMIC_HISTORY

create

update

Process ENTER_ACADEMIC_BACKGROUND

Import Views

View INPUT of entity ACADEMIC_HISTORY

Attributes:

SCHOOL

DEGREE

MAJOR

GPA

DATE

View INPUT of entity STUDENT

Attributes:

SSN

Export Views

View OUTPUT of entity ACADEMIC_HISTORY

Attributes:

SCHOOL

DEGREE

MAJOR

GPA

DATE

View OUTPUT of entity STUDENT

Attributes:

SSN

Entity Action Views

View of entity ACADEMIC_HISTORY

Attributes:

SCHOOL

DEGREE

MAJOR

GPA

DATE

View of entity STUDENT

Attributes:

SSN

Activity Definition

Name: ENTER_DEPENDENT_DATA

Description: This process involves the creation of student's dependent information.

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Expected Effects:

Entity Type

Expected Actions

DEPENDENT

create
update
update
read

STUDENT

Process ENTER_DEPENDENT_DATA

Import Views

View INPUT of entity DEPENDENT

Attributes:

LAST_NAME
FIRST_NAME
FAMILY_MEMBER

View INPUT of entity STUDENT

Attributes:

SSN

Export Views

View OUTPUT of entity DEPENDENT

Attributes:

LAST_NAME FIRST_NAME
FAMILY_MEMBER

View OUTPUT of entity STUDENT

Attributes:

SSN

Entity Action Views

View of entity DEPENDENT

Attributes:

LAST_NAME FIRST_NAME
FAMILY_MEMBER

View of entity STUDENT

Attributes:

SSN

Activity Definition

Name: ESTABLISH_A_NEW_COURSE

Description: System Gen: This process creates a course. (Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COURSE_MAINTENANCE

Expected Effects:	Expected Actions
Entity Type	
-----	-----
COURSE	create

Process ESTABLISH_A_NEW_COURSE

Import Views

View INPUT of entity COURSE

Attributes:

NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
ACADEMIC_DEPARTMENT_CODE
NUMBER

Export Views

View OUTPUT of entity COURSE

Attributes:

NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
ACADEMIC_DEPARTMENT_CODE
NUMBER

Entity Action Views

View of entity COURSE

Attributes:

NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
ACADEMIC_DEPARTMENT_CODE
NUMBER

Activity Definition

Name: ESTABLISH_A_QUARTER_QPR

Description: System Gen: This process involves the creation of an QPR for a particular quarter and year. This process would be called when no QPR exist for that particular quarter and year when a grade is posted. (Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COMPLETED_ACADEMIC_REQUIREMENTS

Expected Effects:

Entity Type

Expected Actions

STUDENT_COURSE_OF_STUDY

update

read

STUDENT

update

read

QUARTER_QPR

create

update

Process ESTABLISH_A_QUARTER_QPR

Import Views

View INPUT of entity QUARTER_QPR

Attributes:

ACADEMIC_YEAR

ACADEMIC_QUARTER

View INPUT of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View IMPORT of entity STUDENT

Attributes:

SSN

Export Views

View EXPORT of entity QUARTER_QPR

Attributes:

ACADEMIC_YEAR

ACADEMIC_QUARTER

Entity Action Views

View of entity QUARTER_QPR

Attributes:

ACADEMIC_YEAR

ACADEMIC_QUARTER

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

View of entity STUDENT

Attributes:

SSN

Activity Definition

Name: ESTABLISH_NEW_CURRICULAR_OFFICE

Description: System Gen: This process creates a new curricular office in addition to the 11 present offices. Additionally, at least one curriculum must be created that composes that particular curricular office. This process requires a modification to the code to add a permitted value for a curricular office code. (Outside scope: Registrar with the assistance of the Curricular Officers and Academic Associates)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: CURRICULUM_OFFICE_MAINTENANCE

Expected Effects:

Entity Type -----	Expected Actions -----
QTR_OF_TYPICAL_STUDY	create update
TYPICAL_COURSE_OF_STUDY	create update
COMPOSITION_OF_TYPICAL_STUDY	create update
CURRICULUM	create update
CURRICULAR_OFFICE	create update
COURSE	update read

Process ESTABLISH_NEW_CURRICULAR_OFFICE

Import Views

Group View GROUP_IMPORT
Cardinality Min: 1 Max: 20 Avg: 5
View INPUT of entity CURRICULUM
Attributes:
TITLE
NUMBER
View INPUT of entity CURRICULAR_OFFICE
Attributes:
TITLE
CODE
PASSWORD

Export Views

Group View GROUP_EXPORT

Cardinality Min: 1 Max: 20 Avg: 5

View OUTPUT of entity CURRICULUM

Attributes:

TITLE

NUMBER

View OUTPUT of entity CURRICULAR_OFFICE

Attributes:

TITLE

CODE

Entity Action Views

View of entity CURRICULUM

Attributes:

TITLE

NUMBER

View of entity CURRICULAR_OFFICE

Attributes:

TITLE

CODE

PASSWORD

Activity Definition

Name: FILE_BOOK_CLAIM

Description: This process creates a claim against a Naval student's book money (total amt they are allowed to spend for an academic year) and reduces the amount remaining in the student's book money.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects: Entity Type -----	Expected Actions -----
STUDENT	read
STUDENT_BOOK_REIMBURSEMENT	update
	read
BOOK_CLAIM	create
	update

Process FILE_BOOK_CLAIM

Import Views

View INPUT of entity BOOK_CLAIM

Attributes:

AMOUNT_OF_CLAIM

ACADEMIC_QUARTER

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR

AMOUNT_REMAINING

Export Views

View OUTPUT of entity BOOK_CLAIM

Attributes:

AMOUNT_OF_CLAIM

ACADEMIC_QUARTER

View OUTPUT of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR

Entity Action Views

View of entity BOOK_CLAIM

Attributes:

AMOUNT OF CLAIM
ACADEMIC_QUARTER

View of subtype NAVY

Attributes:

SSN

View of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR
AMOUNT_REMAINING

Activity Definition

Name: INITIALIZE_STUDENT_RECORD

Description: This function involves the evaluation and preparation for a student to be considered at the Naval Postgraduate School.

Type: Function

Subordinate of: ACADEMIC_DATABASE_ADMINISTRATION

Subordinates: ASSIGN_PROJECTED_STUDENT
MODIFY_PROJECTED_STUDENT
REMOVE_PROJECTED_STUDENT

Activity Definition

Name: MODIFY_ACADEMIC_BACKGROUND

Description: This process involves the modification of a student's record of their academic history.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Expected Effects:	Expected Actions
Entity Type	
-----	-----
ACADEMIC_HISTORY	update read

Process MODIFY_ACADEMIC_BACKGROUND

Import Views

View INPUT_ADJUSTING of entity ACADEMIC_HISTORY
Attributes:
 SCHOOL
 GPA
 DATE

View INPUT of entity STUDENT
Attributes:
 SSN

View PRESENT_INPUT of entity ACADEMIC_HISTORY
Attributes:
 MAJOR
 DEGREE

Export Views

View OUTPUT_ADJUSTED of entity ACADEMIC_HISTORY
Attributes:
 SCHOOL
 GPA
 DATE

View OUTPUT of entity ACADEMIC_HISTORY
Attributes:
 SCHOOL
 DEGREE
 MAJOR
 GPA
 DATE

Entity Action Views

View of entity STUDENT

Attributes:

SSN

View of entity ACADEMIC_HISTORY

Attributes:

SCHOOL

DEGREE

MAJOR

GPA

DATE

Activity Definition

Name: MODIFY_BOOK_CLAIM

Description: This process modifies an existing Naval student's book claim and makes the needed adjustment in the student's book money.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type

Expected Actions

STUDENT_BOOK_REIMBURSEMENT

update

BOOK_CLAIM

read

update

read

Process MODIFY_BOOK_CLAIM

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR

View INPUT_NEW of entity BOOK_CLAIM

Attributes:

ACADEMIC_QUARTER

AMOUNT_OF_CLAIM

Export Views

View OUTPUT of entity BOOK_CLAIM

Attributes:

AMOUNT_OF_CLAIM

ACADEMIC_QUARTER

Entity Action Views

View of subtype NAVY

Attributes:

SSN

View of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR

AMOUNT_REMAINING

View of entity BOOK_CLAIM

Attributes:

AMOUNT_OF_CLAIM

ACADEMIC_QUARTER

Activity Definition

Name: MODIFY_CURRICULAR_OFFICE

Description: System Gen: This process modifies an existing curricular office. (Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: CURRICULUM_OFFICE_MAINTENANCE

Expected Effects:

Entity Type

CURRICULAR_OFFICE

Expected Actions

update
read

Process MODIFY_CURRICULAR_OFFICE

Import Views

View INPUT of entity CURRICULAR_OFFICE

Attributes:

CODE
TITLE

Export Views

View OUTPUT of entity CURRICULAR_OFFICE

Attributes:

TITLE
CODE

Entity Action Views

View of entity CURRICULAR_OFFICE

Attributes:

TITLE
CODE

Activity Definition

Name: MODIFY_CURRICULUM

Description: System Gen: This process involves the modification of a curriculum for a particular curricular office.
(Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: CURRICULUM_OFFICE_MAINTENANCE

Expected Effects:

Entity Type -----	Expected Actions -----
COURSE	read
COMPOSITION_OF_TYPICAL_STUDY	update
TYPICAL_COURSE_OF_STUDY	read
	update
CURRICULUM	read
	update
CURRICULAR_OFFICE	read
	read

Process MODIFY_CURRICULUM

Import Views

View INPUT of entity CURRICULUM
Attributes:
NUMBER
TITLE

Export Views

View OUTPUT of entity CURRICULUM
Attributes:
TITLE
NUMBER

Entity Action Views

View of entity CURRICULUM
Attributes:
TITLE
NUMBER

Activity Definition

Name: MODIFY_DEPENDENT_DATA

Description: This process involves the modification of a student's dependent information.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Expected Effects:

Entity Type

DEPENDENT

Expected Actions

update
read

Process MODIFY_DEPENDENT_DATA

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT of entity DEPENDENT

Attributes:

LAST_NAME

FIRST_NAME

FAMILY_MEMBER

Export Views

View OUTPUT of entity DEPENDENT

Attributes:

LAST_NAME

FIRST_NAME

FAMILY_MEMBER

Entity Action Views

View of entity STUDENT

Attributes:

SSN

View of entity DEPENDENT

Attributes:

LAST_NAME

FIRST_NAME

FAMILY_MEMBER

Activity Definition

Name: MODIFY_EXISTING_COURSE

Description: System Gen: This process modifies an existing course.
(Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COURSE_MAINTENANCE

Expected Effects:

Entity Type

COURSE

Expected Actions

update
read

Process MODIFY_EXISTING_COURSE

Import Views

View INPUT of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE
NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS

Export Views

View OUTPUT of entity COURSE

Attributes:

NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
NUMBER
ACADEMIC_DEPARTMENT_CODE

Entity Action Views

View of entity COURSE

Attributes:

NAME
LECTURE_CREDIT_HOURS
LAB_CREDIT_HOURS
NUMBER
ACADEMIC_DEPARTMENT_CODE

Activity Definition

Name: MODIFY_GRADE

Description: System Gen: This process is a generic update of a student's grade. (Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COMPLETED_ACADEMIC_REQUIREMENTS

Expected Effects:

Entity Type -----	Expected Actions -----
STUDENT_COURSE_OF_STUDY	update read

Process MODIFY_GRADE

Import Views

View INPUT of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

GRADE

View INPUT of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View INPUT of entity STUDENT

Attributes:

SSN

Export Views

View OUTPUT of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

GRADE

Entity Action Views

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity STUDENT

Attributes:

SSN

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

GRADE

Activity Definition

Name: MODIFY_NAVY_BOOK_CEILING

Description: System Gen: This process modifies the Navy Book Eligibility. (Outside scope: MIS)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: SYSTEM_MANAGEMENT

Expected Effects:

Entity Type

Expected Actions

TOTAL_ANNUAL_NAVAL_BOOK_CEILING

update
read

Process MODIFY_NAVY_BOOK_CEILING

Import Views

View INPUT of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING
Attributes:

DATE_IMPLEMENTED
TOTAL_AMOUNT

Export Views

View OUTPUT of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING
Attributes:

TOTAL_AMOUNT
DATE_IMPLEMENTED

Entity Action Views

View of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING
Attributes:

TOTAL_AMOUNT
DATE_IMPLEMENTED

Activity Definition

Name: MODIFY_PASSWORD

Description: System Gen: This process creates a password for a particular curricular office for use by the system to restrict the view available to a given curricular office; the operation of this process will, of course, be transparent to the users. (Outside scope: MIS)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: SYSTEM_MANAGEMENT

Expected Effects:

Entity Type

CURRICULAR_OFFICE

Expected Actions

update

read

Process MODIFY_PASSWORD

Import Views

View INPUT of entity CURRICULAR_OFFICE

Attributes:

CODE

PASSWORD

Export Views

View OUTPUT of entity CURRICULAR_OFFICE

Attributes:

PASSWORD

CODE

Entity Action Views

View of entity CURRICULAR_OFFICE

Attributes:

PASSWORD

CODE

Activity Definition

Name: MODIFY_PROJECTED_STUDENT

Description: This process modifies the description of a prospective student. (Admission or Curricular Officer)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: INITIALIZE_STUDENT_RECORD

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process MODIFY_PROJECTED_STUDENT

Import Views

View INPUT of entity STUDENT

Attributes:

SSN		
opt TYPE_REFRESHER	opt	TYPE_OFFICER
opt RECEIVED_ORDERS_TO_ATTEND	opt	LAST_NAME
opt FIRST_NAME	opt	MIDDLE_INITIAL
opt SHORTNAME	opt	RANK
opt ANTICIPATED_GRADUATION_DATE	opt	APC
opt PROPOSED_NPS_DEGREE	opt	NPS_MAJOR
opt COMMISSIONING_SOURCE	opt	COMMENT1
opt DUAL_DEGREE	opt	PROGRAM
opt COUNTRY	opt	SERVICE
opt INTERNATIONAL_SERVICE_COMPONENT		

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN	TYPE_REFRESHER
PRESENT_STATUS	TYPE_OFFICER
RECEIVED_ORDERS_TO_ATTEND	LAST_NAME
FIRST_NAME	MIDDLE_INITIAL
SHORTNAME	RANK
ANTICIPATED_GRADUATION_DATE	APC
PROPOSED_NPS_DEGREE	NPS_MAJOR
COMMENT1	COMMISSIONING_SOURCE
DUAL_DEGREE	PROGRAM
COUNTRY	SERVICE
INTERNATIONAL_SERVICE_COMPONENT	

Entity Action Views

View of entity STUDENT

Attributes:

SSN	TYPE_REFRESHER
PRESENT_STATUS	TYPE_OFFICER
RECEIVED_ORDERS_TO_ATTEND	LAST_NAME
FIRST_NAME	MIDDLE_INITIAL
SHORTNAME	RANK
ANTICIPATED_GRADUATION_DATE	APC
PROPOSED_NPS_DEGREE	NPS_MAJOR
COMMENT1	COMMISSIONING_SOURCE
DUAL_DEGREE	PROGRAM
COUNTRY	SERVICE
INTERNATIONAL_SERVICE_COMPONENT	

Activity Definition

Name: MODIFY_PRT

Description: This process modifies a Naval Student's Physical Readiness Training record.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type

PRT

Expected Actions

update
read

Process MODIFY_PRT

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT of entity PRT

Attributes:

opt DATE_OF_TEST
opt SCORE
opt BODY_FAT

Export Views

View OUTPUT of entity PRT

Attributes:

DATE_OF_TEST
SCORE
BODY_FAT

Entity Action Views

View of subtype NAVY

Attributes:

SSN

View of entity PRT

Attributes:

DATE_OF_TEST
SCORE
BODY_FAT

Activity Definition

Name: MODIFY_TYPICAL_COURSE_OF_STUDY

Description: This process involves the modification of a catalogued typical course of study for a particular curriculum.

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: COURSE_OF_STUDY_MAINTENANCE

Expected Effects:

Entity Type -----	Expected Actions -----
COMPOSITION_OF_TYPICAL_STUDY	update read
CURRICULUM	read
TYPICAL_COURSE_OF_STUDY	update read
COURSE	update read

Process MODIFY_TYPICAL_COURSE_OF_STUDY

Import Views

View IMPORT_2 of entity COURSE
Attributes:
 NUMBER
 ACADEMIC_DEPARTMENT_CODE

View IMPORT of entity COMPOSITION_OF_TYPICAL_STUDY
Attributes:
 TYPE_OF_COURSE

View IMPORT of entity COURSE
Attributes:
 NUMBER
 ACADEMIC_DEPARTMENT_CODE

View IMPORT of entity CURRICULUM
Attributes:
 NUMBER

View IMPORT of entity TYPICAL_COURSE_OF_STUDY
Attributes:
 REFRESHER_REQUIREMENTS
 TYPE_STUDENT

View IMPORT of entity QTR_OF_TYPICAL_STUDY
Attributes:
 QUARTER_NUMBER

Export Views

View EXPORT_2 of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View EXPORT of entity COURSE

Attributes:

ACADEMIC_DEPARTMENT_CODE

NUMBER

View EXPORT of entity COMPOSITION_OF_TYPICAL_STUDY

Attributes:

TYPE_OF_COURSE

Entity Action Views

View PERSISTENT_3 of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View PERSISTENT_2 of entity COURSE

Attributes:

ACADEMIC_DEPARTMENT_CODE

NUMBER

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity CURRICULUM

Attributes:

NUMBER

View of entity TYPICAL_COURSE_OF_STUDY

Attributes:

REFRESHER_REQUIREMENTS

TYPE_STUDENT

View of entity QTR_OF_TYPICAL_STUDY

Attributes:

QUARTER_NUMBER

View of entity COMPOSITION_OF_TYPICAL_STUDY

Attributes:

TYPE_OF_COURSE

Activity Definition

Name: NAVY_REQUIREMENTS_MAINTENANCE

Description: This function incorporates the maintenance of a Naval Student's Administrative requirements.

Type: Function

Subordinate of: SUPERVISE_ENROLLED_STUDENT

Subordinates:

- RECORD_NAVAL_FITREP
- RECORD_NAVAL_OFFICER_DESCRIPTION
- RECORD_PRT
- MODIFY_PRT
- REMOVE_PRT
- SETUP_NAVAL_BOOK_REIMBURSEMENT
- FILE_BOOK_CLAIM
- MODIFY_BOOK_CLAIM
- REMOVE_BOOK_CLAIM

Activity Definition

Name: PERSONAL_DATA_MAINTENANCE

Description: This function involves the maintenance of information which is not pertinent to the performance of a student at the Naval Postgraduate School.

Type: Function

Subordinate of: SUPERVISE_ENROLLED_STUDENT

Subordinates: ENTER_DEPENDENT_DATA
MODIFY_DEPENDENT_DATA
ELIMINATE_DEPENDENT_DATA
ENTER_ACADEMIC_BACKGROUND
MODIFY_ACADEMIC_BACKGROUND
REMOVE_ACADEMIC_BACKGROUND
RECORD_STUDENT_DATA

Activity Definition

Name: POST_GRADE

Description: System Gen: This process records the grade a student earned at the completion of a course. (Outside scope: Registrar)

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: COMPLETED_ACADEMIC_REQUIREMENTS

Expected Effects:

Entity Type

QUARTER_QPR

STUDENT_COURSE_OF_STUDY

Expected Actions

update
read
update
read

Process POST_GRADE

Import Views

View INPUT of entity STUDENT_COURSE_OF_STUDY
Attributes:

STATUS
ACADEMIC_YEAR
ACADEMIC_QUARTER
GRADE

View INPUT of entity COURSE
Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

View INPUT of entity STUDENT
Attributes:

SSN

Export Views

View OUTPUT of entity STUDENT_COURSE_OF_STUDY
Attributes:

STATUS
ACADEMIC_YEAR
ACADEMIC_QUARTER
GRADE

Entity Action Views

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity STUDENT

Attributes:

SSN

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

ACADEMIC_YEAR

ACADEMIC_QUARTER

GRADE

Activity Definition

Name: RECORD_ARRIVAL

Description: This process involves the recording of the arrival (at the curricular office) of an incoming student. If this student is in the Navy; the Student Book Money is created.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process RECORD_ARRIVAL

Import Views

View ARRIVED of entity STUDENT

Attributes:

SSN
DATE_REPORTED_ABOARD

Export Views

View OUTPUT_ARRIVED of entity STUDENT

Attributes:

SSN
PRESENT_STATUS
RECEIVED_ORDERS_TO_ATTEND
LAST_NAME
FIRST_NAME
MIDDLE_INITIAL
DATE_REPORTED_ABOARD

Entity Action Views

View of entity STUDENT

Attributes:

SSN
PRESENT_STATUS
RECEIVED_ORDERS_TO_ATTEND
LAST_NAME
FIRST_NAME
MIDDLE_INITIAL
DATE_REPORTED_ABOARD

Activity Definition

Name: RECORD_BIRTH_INFO

Description: This process records or updates the date of birth and place of birth of a particular student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process RECORD_BIRTH_INFO

Import Views

View INPUT of entity STUDENT

Attributes:

SSN
DATE_OF_BIRTH
PLACE_OF_BIRTH_CITY
PLACE_OF_BIRTH_STATE

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN
DATE_OF_BIRTH
PLACE_OF_BIRTH_CITY
PLACE_OF_BIRTH_STATE

Entity Action Views

View of entity STUDENT

Attributes:

SSN
DATE_OF_BIRTH
PLACE_OF_BIRTH_CITY
PLACE_OF_BIRTH_STATE

Activity Definition

Name: RECORD_DESCRIPTIVE_INFO

Description: This process updates descriptive information of a particular student. (ex. name, gender, rank, commissioning source, etc.)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process RECORD_DESCRIPTIVE_INFO

Import Views

View INPUT of entity STUDENT

Attributes:

SSN	
opt TYPE_REFRESHER	opt PRESENT_STATUS
opt TYPE_OFFICER	opt RECEIVED_ORDERS_TO_ATTEND
opt LAST_NAME	opt FIRST_NAME
opt MIDDLE_INITIAL	opt SHORTNAME
opt GENDER	opt RANK
opt DATE_OF_RANK	opt MARITAL_STATUS
opt COMMISSIONING_SOURCE	opt LIBRARY_CARD_NUMBER
opt MAINFRAME_ACCOUNT_NUMBER	

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN	TYPE_REFRESHER
PRESENT_STATUS	TYPE_OFFICER
LAST_NAME	RECEIVED_ORDERS_TO_ATTEND
FIRST_NAME	MIDDLE_INITIAL
SHORTNAME	GENDER
RANK	DATE_OF_RANK
MARITAL_STATUS	COMMISSIONING_SOURCE
LIBRARY_CARD_NUMBER	MAINFRAME_ACCOUNT_NUMBER

Entity Action Views

View of entity STUDENT

Attributes:

SSN	TYPE_REFRESHER
PRESENT_STATUS	TYPE_OFFICER
LAST_NAME	RECEIVED_ORDERS_TO_ATTEND
FIRST_NAME	MIDDLE_INITIAL
SHORTNAME	GENDER
RANK	DATE_OF_RANK
MARITAL_STATUS	COMMISSIONING_SOURCE
LIBRARY_CARD_NUMBER	MAINFRAME_ACCOUNT_NUMBER

Activity Definition

Name: RECORD_LOCATION_INFO

Description: This process records or updates the local address, phone number, section number, study space, or locker number of a particular student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process RECORD_LOCATION_INFO

Import Views

View INPUT of entity STUDENT

Attributes:

SSN	PHONE_NUMBER
STREET	CITY
ZIP_CODE	LAMESA_HOUSING_OCCUPANT
SMC_BOX_NUMBER	SECTION_NUMBER
SPLIT_SECTION	STUDY_SPACE
LOCKER_NUMBER	

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN	PHONE_NUMBER
STREET	CITY
ZIP_CODE	LAMESA_HOUSING_OCCUPANT
SMC_BOX_NUMBER	SECTION_NUMBER
SPLIT_SECTION	STUDY_SPACE
LOCKER_NUMBER	

Entity Action Views

View of entity STUDENT

Attributes:

SSN	PHONE_NUMBER
STREET	CITY
ZIP_CODE	LAMESA_HOUSING_OCCUPANT
SMC_BOX_NUMBER	SECTION_NUMBER
SPLIT_SECTION	STUDY_SPACE
LOCKER_NUMBER	

Activity Definition

Name: RECORD_MEDICAL_INFO

Description: This process updates the medical information maintained on a particular student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process RECORD_MEDICAL_INFO

Import Views

View INPUT of entity STUDENT

Attributes:

SSN
AIDS_TEST_DATE
PHYSICAL_DATE
DENTAL_DATE

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN
AIDS_TEST_DATE
PHYSICAL_DATE
DENTAL_DATE

Entity Action Views

View of entity STUDENT

Attributes:

SSN
AIDS_TEST_DATE
PHYSICAL_DATE
DENTAL_DATE

Activity Definition

Name: RECORD_NAVAL_FITREP

Description: This process records the date that a Naval Fitness Report was submitted, and the due date of the next report.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type

STUDENT

Expected Actions

update

read

Process RECORD_NAVAL_FITREP

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

LAST_FITREP_DATE

NEXT_FITREP_DUE

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN

LAST_FITREP_DATE

NEXT_FITREP_DUE

Entity Action Views

View of entity STUDENT

Attributes:

SSN

LAST_FITREP_DATE

NEXT_FITREP_DUE

Activity Definition

Name: RECORD_NAVAL_OFFICER_DESCRIPTION

Description: This process involves the recording of a Naval Officer's Lineal-Number, Year-Group, and Officer-Designator.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type

STUDENT

Expected Actions

update

read

Process RECORD_NAVAL_OFFICER_DESCRIPTION

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

TYPE OFFICER

opt OFFICER DESIGNATOR

opt LINEAL NUMBER

opt OFFICER_YEAR_GROUP

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN

TYPE OFFICER

OFFICER DESIGNATOR

LINEAL NUMBER

OFFICER_YEAR_GROUP

Entity Action Views

View of entity STUDENT

Attributes:

SSN

TYPE OFFICER

OFFICER DESIGNATOR

LINEAL NUMBER

OFFICER_YEAR_GROUP

Activity Definition

Name: RECORD_PRT

Description: This process involves the creation of a Naval Student's Physical Readiness Training Record.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type	Expected Actions
-----	-----
NAVY	update
STUDENT	read
PRT	create
	update

Process RECORD_PRT

Import Views

View INPUT of entity PRT
Attributes:
SCORE
opt BODY_FAT DATE_OF_TEST

View INPUT of entity STUDENT
Attributes:
SSN

Export Views

View OUTPUT of entity PRT
Attributes:
SCORE
BODY_FAT DATE_OF_TEST

View OUTPUT of entity STUDENT
Attributes:
SSN

Entity Action Views

View of entity PRT
Attributes:
SCORE
BODY_FAT DATE_OF_TEST

View of entity STUDENT
Attributes:
SSN

Activity Definition

Name: RECORD_SECURITY_INFO

Description: This process records the security attributes of a particular student. (ex. background, access)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type -----	Expected Actions -----
STUDENT	update read

Process RECORD_SECURITY_INFO

Import Views
View INPUT of entity STUDENT
Attributes:
SSN
SECURITY_BACKGROUND
SECURITY_ACCESS

Export Views
View OUTPUT of entity STUDENT
Attributes:
SSN
SECURITY_BACKGROUND
SECURITY_ACCESS

Entity Action Views
View of entity STUDENT
Attributes:
SSN
SECURITY_BACKGROUND
SECURITY_ACCESS

Activity Definition

Name: RECORD_STATION_INFO

Description: This process records the previous duty station, next duty station and date of orders of a particular student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type

STUDENT

Expected Actions

update

read

Process RECORD_STATION_INFO

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

PREVIOUS_DUTY_STATION

NEXT_DUTY_STATION

DATE_OF_ORDERS

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN

PREVIOUS_DUTY_STATION

NEXT_DUTY_STATION

DATE_OF_ORDERS

Entity Action Views

View of entity STUDENT

Attributes:

SSN

PREVIOUS_DUTY_STATION

NEXT_DUTY_STATION

DATE_OF_ORDERS

Activity Definition

Name: RECORD_STUDENT_DATA

Description: This process involves the updating of non-academic information about a student such as address, phone number, birthdate, etc.

Type: Process
Not Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Subordinates: RECORD_MEDICAL_INFO
RECORD_LOCATION_INFO
RECORD_BIRTH_INFO
RECORD_STATION_INFO
RECORD_DESCRIPTIVE_INFO
RECORD_SECURITY_INFO
UPDATE_DEGREE_INFO

Expected Effects:

Entity Type

CURRICULUM
STUDENT

Expected Actions

read
update
read

Activity Definition

Name: RECORD_THESIS_PROPOSAL

Description: System Gen: This process records the creation of a thesis for a particular student or students. (Outside Scope: Thesis Processor)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type

STUDENT

THESIS

Expected Actions

update

read

create

update

Process RECORD_THESIS_PROPOSAL

Import Views

View INDICATED of entity THESIS

Attributes:

DUE_DATE_YEAR

STATUS

SECOND_READER

JOINT

DUE_DATE_MONTH

TITLE

ADVISOR

CLASSIFIED

NUMBER

View IDENTIFIED of entity STUDENT

Attributes:

SSN

Export Views

View OUTPUT of entity THESIS

Attributes:

DUE_DATE_YEAR

STATUS

SECOND_READER

JOINT

DUE_DATE_MONTH

TITLE

ADVISOR

CLASSIFIED

NUMBER

View EXPORT of entity STUDENT

Attributes:

SSN

Entity Action Views

View of entity THESIS

Attributes:

DUE_DATE_YEAR

STATUS

SECOND_READER

JOINT

DUE_DATE_MONTH

View of entity STUDENT

Attributes:

SSN

TITLE

ADVISOR

CLASSIFIED

NUMBER

Activity Definition

Name: REMOVE_ACADEMIC_BACKGROUND

Description: This process removes an erroneously entered student academic history.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: PERSONAL_DATA_MAINTENANCE

Expected Effects:

Entity Type

ACADEMIC_HISTORY

Expected Actions

read
delete

Process REMOVE_ACADEMIC_BACKGROUND

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT_REMOVING of entity ACADEMIC_HISTORY

Attributes:

MAJOR

DEGREE

Export Views

View OUTPUT_REMOVED of entity ACADEMIC_HISTORY

Attributes:

DEGREE

MAJOR

Entity Action Views

View of entity STUDENT

Attributes:

SSN

View of entity ACADEMIC_HISTORY

Attributes:

DEGREE

MAJOR

Activity Definition

Name: REMOVE_BOOK_CLAIM

Description: This process removes an erroneously entered book claim.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type

Expected Actions

BOOK_CLAIM

read
delete
update
read

STUDENT_BOOK_REIMBURSEMENT

Process REMOVE_BOOK_CLAIM

Import Views

View INPUT of entity STUDENT

Attributes:

SSN

View INPUT of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR

View INPUT of entity BOOK_CLAIM

Attributes:

ACADEMIC_QUARTER

Export Views

View OUTPUT of entity BOOK_CLAIM

Attributes:

AMOUNT_OF_CLAIM

ACADEMIC_QUARTER

Entity Action Views

View of subtype NAVY

Attributes:

SSN

View of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

YEAR

View of entity BOOK_CLAIM

Attributes:

AMOUNT_OF_CLAIM

ACADEMIC_QUARTER

Activity Definition

Name: REMOVE_COURSE_FROM_CATALOG

Description: System Gen: This process removes a course from the available course listing. (Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COURSE_MAINTENANCE

Expected Effects:

Entity Type

COURSE

Expected Actions

read
delete

Process REMOVE_COURSE_FROM_CATALOG

Import Views

View INPUT of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

Export Views

View OUTPUT of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

Entity Action Views

View of entity COURSE

Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

Activity Definition

Name: REMOVE_CURRICULAR_OFFICE

Description: System Gen: This process removes a curricular office.
(Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: CURRICULUM_OFFICE_MAINTENANCE

Expected Effects:

Entity Type	Expected Actions
-----	-----
CURRICULAR_OFFICE	read delete

Process REMOVE_CURRICULAR_OFFICE

Import Views

View INPUT of entity CURRICULAR_OFFICE

Attributes:

CODE

Export Views

View OUTPUT of entity CURRICULAR_OFFICE

Attributes:

CODE

Entity Action Views

View of entity CURRICULAR_OFFICE

Attributes:

CODE

Activity Definition

Name: REMOVE_ERRONEOUS_THESIS

Description: System Gen: This process removes a thesis which has been erroneously entered or has been abandoned by its author, as distinguished from a mere modification.
(Outside scope: Thesis Processor)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type

STUDENT
THESIS

Expected Actions

read
read
delete

Process REMOVE_ERRONEOUS_THESIS

Import Views

View INPUT of entity THESIS

Attributes:

DUE_DATE_YEAR
NUMBER

Export Views

View OUTPUT of entity THESIS

Attributes:

NUMBER
DUE_DATE_YEAR

Entity Action Views

View of entity THESIS

Attributes:

NUMBER
DUE_DATE_YEAR

Activity Definition

Name: REMOVE_GRADUATES

Description: System Gen: This process archives the records of those students who have graduated (or those who attended but did not meet the requirements for graduation) from the Naval Postgraduate School. (Outside scope: Registrar)

Procedure actually accomplished by Registrar ONLY, however, must be modeled here to enable view capability by curricular officers.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COMPLETED_ACADEMIC_REQUIREMENTS

Expected Effects:

Entity Type

Expected Actions

THESIS

read
delete

STUDENT

read
delete

Process REMOVE_GRADUATES

Import Views

View INPUT of entity STUDENT
Attributes: SSN

Export Views

View OUTPUT of entity STUDENT
Attributes: SSN
ANTICIPATED_GRADUATION_DATE

Entity Action Views

View of entity STUDENT
Attributes: SSN
ANTICIPATED_GRADUATION_DATE

Activity Definition

Name: REMOVE_PROJECTED_STUDENT

Description: System Gen: This process removes an erroneously entered student. (Outside Scope: Admissions)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: INITIALIZE_STUDENT_RECORD

Expected Effects:

Entity Type

STUDENT

Expected Actions

read
delete

Process REMOVE_PROJECTED_STUDENT

Import Views

View INPUT REMOVING of entity STUDENT

Attributes:

SSN

Export Views

View REMOVED of entity STUDENT

Attributes:

SSN

PRESENT_STATUS

LAST_NAME

FIRST_NAME

RANK

TYPE_REFRESHER

TYPE_OFFICER

RECEIVED_ORDERS_TO_ATTEND

MIDDLE_INITIAL

ANTICIPATED_GRADUATION_DATE

Entity Action Views

View of entity STUDENT

Attributes:

SSN

PRESENT_STATUS

LAST_NAME

FIRST_NAME

RANK

TYPE_REFRESHER

TYPE_OFFICER

RECEIVED_ORDERS_TO_ATTEND

MIDDLE_INITIAL

ANTICIPATED_GRADUATION_DATE

Activity Definition

Name: REMOVE_PRT

Description: This process removes an erroneously entered PRT.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type

PRT

Expected Actions

read
delete

Process REMOVE_PRT

Import Views

View INPUT of entity STUDENT
Attributes:

SSN

View INPUT of entity PRT

Attributes:

DATE_OF_TEST

Export Views

View OUTPUT of entity PRT

Attributes:

DATE_OF_TEST

SCORE

BODY_FAT

Entity Action Views

View of subtype NAVY

Attributes:

SSN

View of entity PRT

Attributes:

DATE_OF_TEST

SCORE

BODY_FAT

Activity Definition

Name: REMOVE_TYPICAL_COURSE_OF_STUDY

Description: This process removes an erroneously entered typical course of study.

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: COURSE_OF_STUDY_MAINTENANCE

Expected Effects:

Entity Type -----	Expected Actions -----
TYPICAL_COURSE_OF_STUDY	read delete

Process REMOVE_TYPICAL_COURSE_OF_STUDY

Import Views

View INPUT of entity CURRICULUM
Attributes:
NUMBER

View INPUT of entity TYPICAL_COURSE_OF_STUDY
Attributes:
REFRESHER_REQUIREMENTS
TYPE_STUDENT

Export Views

View OUTPUT of entity TYPICAL_COURSE_OF_STUDY
Attributes:
REFRESHER_REQUIREMENTS
TYPE_STUDENT

Entity Action Views

View of entity CURRICULUM
Attributes:
NUMBER

View of entity TYPICAL_COURSE_OF_STUDY
Attributes:
REFRESHER_REQUIREMENTS
TYPE_STUDENT

Activity Definition

Name: REVISE_THESIS_PROPOSAL

Description: System Gen: This process modifies an existing thesis.
If a student must be removed from a joint thesis, a
disassociation would be required. (Outside scope:
Thesis Processor)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type -----	Expected Actions -----
THESIS	update read
STUDENT	update read

Process REVISE_THESIS_PROPOSAL

Import Views

View INPUT of entity THESIS

Attributes:

DUE_DATE_YEAR	NUMBER
TITLE	STATUS
ADVISOR	SECOND_READER
CLASSIFIED	

Export Views

View OUTPUT of entity THESIS

Attributes:

DUE_DATE_YEAR	NUMBER
TITLE	STATUS
ADVISOR	SECOND_READER
CLASSIFIED	JOINT

Entity Action Views

View of entity THESIS

Attributes:

DUE_DATE_YEAR	NUMBER
TITLE	STATUS
ADVISOR	SECOND_READER
CLASSIFIED	JOINT

Activity Definition

Name: REVISE_THESIS_TO_JOINT_STATUS

Description: System Gen: This process revises a thesis entity to reflect a joint status and associates the thesis with an additional student. (Outside scope: Thesis Processor)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type -----	Expected Actions -----
THESIS	update read
STUDENT	update read

Process REVISE_THESIS_TO_JOINT_STATUS

Import Views

View INPUT_ADDITIONAL of entity STUDENT
Attributes:
SSN

View INPUT of entity THESIS
Attributes:
DUE_DATE_YEAR NUMBER

Export Views

View OUTPUT_ADDITIONAL of entity STUDENT
Attributes:
SSN

View OUTPUT of entity THESIS
Attributes:
DUE_DATE_YEAR NUMBER
JOINT

Entity Action Views

View of entity STUDENT
Attributes:
SSN

View of entity THESIS
Attributes:
DUE_DATE_YEAR NUMBER
JOINT

Activity Definition

Name: SCHEDULE_COURSE

Description: System Gen: This process involves scheduling a course requested by a specific student. (Outside scope: Registrar)

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: COMPLETED_ACADEMIC_REQUIREMENTS

Expected Effects:	Expected Actions
Entity Type	
-----	-----
STUDENT_COURSE_OF_STUDY	update read

Process SCHEDULE_COURSE

Import Views

View INPUT of entity STUDENT_COURSE_OF_STUDY
Attributes:

STATUS
SECTION_NUMBER

View INPUT of entity COURSE
Attributes:

NUMBER
ACADEMIC_DEPARTMENT_CODE

View INPUT of entity STUDENT
Attributes:

SSN

Export Views

View OUTPUT of entity STUDENT_COURSE_OF_STUDY
Attributes:

STATUS
ACADEMIC_YEAR
ACADEMIC_QUARTER
SECTION_NUMBER

Entity Action Views

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity STUDENT

Attributes:

SSN

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

ACADEMIC_YEAR

ACADEMIC_QUARTER

SECTION_NUMBER

Activity Definition

Name: SEND_SPONSOR_LETTER

Description: An incoming student's files will reflect when the student sponsor sent his introductory letter to the new student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COUNSELING_FUTURE_STUDENT

Expected Effects:

Entity Type

STUDENT

Expected Actions

update

read

Process SEND_SPONSOR_LETTER

Import Views

View INCOMING of entity STUDENT

Attributes:

SSN

DATE_SPONSOR_LETTER_SENT

Export Views

View OUTPUT_INCOMING of entity STUDENT

Attributes:

SSN

LAST_NAME

FIRST_NAME

MIDDLE_INITIAL

DATE_SPONSOR_LETTER_SENT

Entity Action Views

View of entity STUDENT

Attributes:

SSN

LAST_NAME

FIRST_NAME

MIDDLE_INITIAL

DATE_SPONSOR_LETTER_SENT

Activity Definition

Name: SEND_WELCOME_ABOARD_PACKAGE

Description: The incoming student's files will reflect when a welcome aboard package was sent to the new student.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: COUNSELING_FUTURE_STUDENT

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process SEND_WELCOME_ABOARD_PACKAGE

Import Views

View INCOMING of entity STUDENT

Attributes:

SSN
DATE_WELCOME_PACKAGE_SENT

Export Views

View OUTPUT of entity STUDENT

Attributes:

SSN
LAST_NAME
FIRST_NAME
MIDDLE_INITIAL
DATE_WELCOME_PACKAGE_SENT

Entity Action Views

View of entity STUDENT

Attributes:

SSN
LAST_NAME
FIRST_NAME
MIDDLE_INITIAL
DATE_WELCOME_PACKAGE_SENT

Activity Definition

Name: SETUP_NAVAL_BOOK_REIMBURSEMENT

Description: This process creates a student's allocated book money for an academic year. (prorated based on the number of quarters remaining in the academic year or on the time a student entered in the academic year).

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: NAVY_REQUIREMENTS_MAINTENANCE

Expected Effects:

Entity Type -----	Expected Actions -----
NAVY	update
TOTAL_ANNUAL_NAVAL_BOOK_CEILING	update
	read
STUDENT	read
STUDENT_BOOK_REIMBURSEMENT	create
	update

Process SETUP_NAVAL_BOOK_REIMBURSEMENT

Import Views

View INPUT of entity STUDENT_BOOK_REIMBURSEMENT
Attributes:

YEAR

View INPUT of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING
Attributes:

DATE_IMPLEMENTED

View INPUT of entity STUDENT
Attributes:

SSN

Export Views

View OUTPUT of entity STUDENT_BOOK_REIMBURSEMENT
Attributes:

AMOUNT_REMAINING

TOTAL_AMOUNT_ELIGIBLE

NUMBER_ACADEMIC_QTRS_AUTHORIZED

YEAR

Entity Action Views

View of entity STUDENT_BOOK_REIMBURSEMENT

Attributes:

AMOUNT_REMAINING
TOTAL_AMOUNT_ELIGIBLE
NUMBER_ACADEMIC_QTRS_AUTHORIZED
YEAR

View of entity TOTAL_ANNUAL_NAVAL_BOOK_CEILING

Attributes:

DATE_IMPLEMENTED
TOTAL_AMOUNT

View of entity STUDENT

Attributes:

SSN

Activity Definition

Name: SETUP_NEW_CURRICULUM

Description: System Gen: This process involves the creation of a new curriculum for a particular curricular office.
(Outside scope: Registrar)

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: CURRICULUM_OFFICE_MAINTENANCE

Expected Effects: Entity Type -----	Expected Actions -----
TYPICAL_COURSE_OF_STUDY	create
COURSE	read
COMPOSITION_OF_TYPICAL_STUDY	create
CURRICULAR_OFFICE	update
	read
CURRICULUM	create
	update

Process SETUP_NEW_CURRICULUM

Import Views

View INPUT of entity CURRICULUM
Attributes:
TITLE NUMBER
View INPUT of entity CURRICULAR_OFFICE
Attributes:
CODE

Export Views

View OUTPUT of entity CURRICULUM
Attributes:
TITLE NUMBER
View OUTPUT of entity CURRICULAR_OFFICE
Attributes:
CODE

Entity Action Views

View of entity CURRICULUM
Attributes:
TITLE NUMBER
View of entity CURRICULAR_OFFICE
Attributes:
CODE

Activity Definition

Name: SETUP_STUDENT_COURSE_OF_STUDY

Description: This process involves the creation of a student's request for all courses he/she will need at NPS. The typical course of study for the student's curriculum will be used as a guide.

Type: Elementary process
Repetitive
Online implementation suggested

Subordinate of: ACADEMIC_COUNSELING

Expected Effects:

Entity Type	Expected Actions
-----	-----
QTR_OF_TYPICAL_STUDY	read
STUDENT	update
	read
COURSE	update
	read
COMPOSITION_OF_TYPICAL_STUDY	read
STUDENT_COURSE_OF_STUDY	create
	update
TYPICAL_COURSE_OF_STUDY	read

Process SETUP_STUDENT_COURSE_OF_STUDY

Import Views

View INPUT of entity STUDENT
Attributes:
SSN

Export Views

Group View OUTGROUP_STUDY
Cardinality Min: 1 Max: 14 Avg: 7
View PROVIDED of entity STUDENT_COURSE_OF_STUDY
Attributes:
STATUS VALIDATION
PASS_FAIL ACADEMIC_YEAR
ACADEMIC_QUARTER

View OUTPUT of entity STUDENT
Attributes:
SSN

View OUTPUT of entity COURSE
Attributes:
NUMBER ACADEMIC_DEPARTMENT_CODE

Local Views

View TEMP of work group IEF_SUPPLIED

Attributes:

COUNT

View TEMP of entity STUDENT_COURSE_OF_STUDY

Attributes:

ACADEMIC_YEAR

ACADEMIC_QUARTER

Entity Action Views

View of entity CURRICULUM

Attributes:

NUMBER

View of entity TYPICAL_COURSE_OF_STUDY

Attributes:

REFRESHER_REQUIREMENTS

TYPE_STUDENT

View of entity QTR_OF_TYPICAL_STUDY

Attributes:

QUARTER_NUMBER

View of entity COMPOSITION_OF_TYPICAL_STUDY

Attributes:

TYPE_OF_COURSE

View of entity STUDENT_COURSE_OF_STUDY

Attributes:

STATUS

VALIDATION

PASS_FAIL

ACADEMIC_YEAR

ACADEMIC_QUARTER

View of entity STUDENT

Attributes:

SSN

TYPE_REFRESHER

TYPE_OFFICER

CONVENING_DATE

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

Activity Definition

Name: SETUP_TYPICAL_COURSE_OF_STUDY

Description: This process involves the creation of a catalog identification of a new typical course of study for a particular curriculum.

Type: Elementary process
 Repetitive
 Online implementation suggested

Subordinate of: COURSE_OF_STUDY_MAINTENANCE

Expected Effects:

Entity Type -----	Expected Actions -----
QTR_OF_TYPICAL_STUDY	create
	update
COMPOSITION_OF_TYPICAL_STUDY	create
	update
TYPICAL_COURSE_OF_STUDY	create
	update
CURRICULUM	update
	read
COURSE	update
	read

Process SETUP_TYPICAL_COURSE_OF_STUDY

Import Views

Group View GROUP_IMPORT
 Cardinality Min: 1 Max: 12 Avg: 6

Group View GROUP_IMPORT_2
 Cardinality Min: 1 Max: 8 Avg: 4

View INPUT of entity COMPOSITION_OF_TYPICAL_STUDY
 Attributes:
 TYPE_OF_COURSE

View INPUT of entity COURSE
 Attributes:
 NUMBER ACADEMIC_DEPARTMENT_CODE

View INPUT of entity QTR_OF_TYPICAL_STUDY
 Attributes:
 QUARTER_NUMBER

View INPUT of entity TYPICAL_COURSE_OF_STUDY
 Attributes:
 TYPE_STUDENT REFRESHER_REQUIREMENTS

View INPUT of entity CURRICULUM
 Attributes:
 NUMBER

Export Views

Group View GROUP_EXPORT

Cardinality Min: 1 Max: 12 Avg: 6

Group View GROUP_EXPORT_2

Cardinality Min: 1 Max: 8 Avg: 4

View OUTPUT of entity COMPOSITION_OF_TYPICAL_STUDY

Attributes:

TYPE_OF_COURSE

View OUTPUT of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View OUTPUT of entity QTR_OF_TYPICAL_STUDY

Attributes:

QUARTER_NUMBER

View OUTPUT of entity TYPICAL_COURSE_OF_STUDY

Attributes:

TYPE_STUDENT

REFRESHER_REQUIREMENTS

View OUTPUT of entity CURRICULUM

Attributes:

NUMBER

Entity Action Views

View of entity COMPOSITION_OF_TYPICAL_STUDY

Attributes:

TYPE_OF_COURSE

View of entity COURSE

Attributes:

NUMBER

ACADEMIC_DEPARTMENT_CODE

View of entity QTR_OF_TYPICAL_STUDY

Attributes:

QUARTER_NUMBER

View of entity TYPICAL_COURSE_OF_STUDY

Attributes:

TYPE_STUDENT

REFRESHER_REQUIREMENTS

View of entity CURRICULUM

Attributes:

NUMBER

Activity Definition

Name: SUPERVISE_ENROLLED_STUDENT

Description: This function includes the maintenance of the academic records of those students enrolled at the Naval Postgraduate School under the supervision of the Curricular Officer.

Type: Function

Subordinate of: ACADEMIC_DATABASE_ADMINISTRATION

Subordinates: ACADEMIC_COUNSELING
PERSONAL_DATA_MAINTENANCE
NAVY_REQUIREMENTS_MAINTENANCE

Activity Definition

Name: SYSTEM_MANAGEMENT

Description: This function incorporates the management of the curricular officer passwords and the ceiling limit of the Navy Book Eligibility. These areas are the responsibility of specially authorized system managers only.

Type: Function

Subordinate of: ACADEMIC_DATABASE_ADMINISTRATION

Subordinates: MODIFY_PASSWORD
ASSIGN_NAVY_BOOK_CEILING
MODIFY_NAVY_BOOK_CEILING

Activity Definition

Name: UPDATE_DEGREE_INFO

Description: This process updates a student's major, type degree, accreditation, or dual degree status.

Type: Elementary process
Not Repetitive
Online implementation suggested

Subordinate of: RECORD_STUDENT_DATA

Expected Effects:

Entity Type

STUDENT

Expected Actions

update
read

Process UPDATE_DEGREE_INFO

Import Views

View INPUT of entity STUDENT

Attributes:

	SSN
opt	PROPOSED_NPS_DEGREE
opt	ACCREDITATION_STATUS
opt	NPS_MAJOR
opt	DUAL_DEGREE

Export Views

View OUTPUT of entity STUDENT

Attributes:

	SSN
	PROPOSED_NPS_DEGREE
	ACCREDITATION_STATUS
	NPS_MAJOR
	DUAL_DEGREE

Entity Action Views

View of entity STUDENT

Attributes:

	SSN
	PROPOSED_NPS_DEGREE
	ACCREDITATION_STATUS
	NPS_MAJOR
	DUAL_DEGREE

-End of Report-

APPENDIX G

The report on the following pages, defines the Activity Hierarchy (or Process Hierarchy as referenced in earlier versions of IEF) of the designed system which shows the hierarchy of operations in the activity model.

[Ref. 27:p. 32-11]

Functions or high-level processes are groups of business activities that together completely support one aspect of furthering the mission of the enterprise. Each function describes something the business does, while an elementary process (the smallest unit of activity in a business) performs a creation, update or deletion of some attribute(s).

Activity Hierarchy

Function	1	ACADEMIC DATABASE ADMINISTRATION
Function	2	1 INITIALIZE STUDENT RECORD
Elem Proc	3	1.1 ASSIGN PROJECTED STUDENT
Elem Proc	3	1.2 MODIFY PROJECTED STUDENT
Elem Proc	3	1.3 REMOVE PROJECTED STUDENT
Function	2	2 COUNSELING FUTURE STUDENT
Elem Proc	3	2.1 ASSIGN STUDENT SPONSOR
Elem Proc	3	2.2 SEND SPONSOR LETTER
Elem Proc	3	2.3 SEND WELCOME ABOARD PACKAGE
Function	2	3 SUPERVISE ENROLLED STUDENT
Function	3	3.1 ACADEMIC COUNSELING
Elem Proc	4	3.1.1 RECORD ARRIVAL
Elem Proc	4	3.1.2 SETUP STUDENT COURSE OF STUDY
Elem Proc	4	3.1.3 CHANGE COURSE IN STUDENT STUDY
Elem Proc	4	3.1.4 CHANGE REQ OF COURSE REQUEST
Elem Proc	4	3.1.5 RECORD THESIS PROPOSAL
Elem Proc	4	3.1.6 REVISE THESIS PROPOSAL
Elem Proc	4	3.1.7 ELIMINATE JOINT STATUS
Elem Proc	4	3.1.8 REVISE THESIS TO JOINT STATUS
Elem Proc	4	3.1.9 ARCHIVE THESIS
Elem Proc	4	3.1.10 REMOVE ERRONEOUS THESIS
Function	3	3.2 PERSONAL DATA MAINTENANCE
Elem Proc	4	3.2.1 ENTER DEPENDENT DATA
Elem Proc	4	3.2.2 MODIFY DEPENDENT DATA
Elem Proc	4	3.2.3 ELIMINATE DEPENDENT DATA
Elem Proc	4	3.2.4 ENTER ACADEMIC BACKGROUND
Elem Proc	4	3.2.5 MODIFY ACADEMIC BACKGROUND
Elem Proc	4	3.2.6 REMOVE ACADEMIC BACKGROUND
Process	4	3.2.7 RECORD STUDENT DATA
Elem Proc	5	3.2.7.1 RECORD MEDICAL INFO
Elem Proc	5	3.2.7.2 RECORD LOCATION INFO
Elem Proc	5	3.2.7.3 RECORD BIRTH INFO
Elem Proc	5	3.2.7.4 RECORD STATION INFO
Elem Proc	5	3.2.7.5 RECORD DESCRIPTIVE INFO
Elem Proc	5	3.2.7.6 RECORD SECURITY INFO
Elem Proc	5	3.2.7.7 UPDATE DEGREE INFO
Function	3	3.3 NAVY REQUIREMENTS MAINTENANCE
Elem Proc	4	3.3.1 RECORD NAVAL FITREP
Elem Proc	4	3.3.2 RECORD NAVAL OFFICER DESCRIPTION
Elem Proc	4	3.3.3 RECORD PRT
Elem Proc	4	3.3.4 MODIFY PRT
Elem Proc	4	3.3.5 REMOVE PRT
Elem Proc	4	3.3.6 SETUP NAVAL BOOK REIMBURSEMENT
Elem Proc	4	3.3.7 FILE BOOK CLAIM
Elem Proc	4	3.3.8 MODIFY BOOK CLAIM
Elem Proc	4	3.3.9 REMOVE BOOK CLAIM

Activity Hierarchy

Function	2	4	COMPLETED ACADEMIC REQUIREMENTS
Elem Proc	3	4.1	SCHEDULE COURSE
Elem Proc	3	4.2	POST GRADE
Elem Proc	3	4.3	ESTABLISH A QUARTER QPR
Elem Proc	3	4.4	REMOVE GRADUATES
Elem Proc	3	4.5	MODIFY GRADE
Function	2	5	CURRICULUM DEVELOPMENT MGMT
Function	3	5.1	CURRICULUM OFFICE MAINTENANCE
Elem Proc	4	5.1.1	ESTABLISH NEW CURRICULAR OFFICE
Elem Proc	4	5.1.2	MODIFY CURRICULAR OFFICE
Elem Proc	4	5.1.3	REMOVE CURRICULAR OFFICE
Elem Proc	4	5.1.4	SETUP NEW CURRICULUM
Elem Proc	4	5.1.5	MODIFY CURRICULUM
Elem Proc	4	5.1.6	ELIMINATE CURRICULUM
Function	3	5.2	COURSE OF STUDY MAINTENANCE
Elem Proc	4	5.2.1	SETUP TYPICAL COURSE OF STUDY
Elem Proc	4	5.2.2	MODIFY TYPICAL COURSE OF STUDY
Elem Proc	4	5.2.3	REMOVE TYPICAL COURSE OF STUDY
Function	2	6	COURSE MAINTENANCE
Elem Proc	3	6.1	ESTABLISH A NEW COURSE
Elem Proc	3	6.2	MODIFY EXISTING COURSE
Elem Proc	3	6.3	REMOVE COURSE FROM CATALOG
Function	2	7	SYSTEM MANAGEMENT
Elem Proc	3	7.1	MODIFY PASSWORD
Elem Proc	3	7.2	ASSIGN NAVY BOOK CEILING
Elem Proc	3	7.3	MODIFY NAVY BOOK CEILING

-End of Report-

APPENDIX H

The Action Diagrams on the following pages were produced during system analysis and represents the views of a process. These diagrams depict the logic of a process in terms of the actions carried out on each entity involved and the conditions constraining these actions.

The smallest unit of activity that is meaningful to the end user in the business and leaves the business in a consistent state is the elementary process. For every elementary process, there is an associated Action Diagram. [Ref. 28:p. 8-14]

Although many of the processes indicated are not under the Curricular Officers' cognizance, each attribute within a developed data model must have a process which creates, updates, or deletes it (unless unique to the business function). Since the Curricular Officers' Business Area must be given a read access to these attributes, processes were developed and described as System Gen in the process descriptions of the Action Diagrams.

Procedures detailing how these processes will be implemented must be developed in follow-on actions. At that time screen designs can be produced which lay out the fields, literals, and special attributes of an import or export view.

The outline of elementary processes below is described in the Action Diagrams on the following pages:

- 1.1 ASSIGN_PROJECTED_STUDENT
- 1.2 MODIFY_PROJECTED_STUDENT
- 1.3 REMOVE_PROJECTED_STUDENT

- 2.1 ASSIGN_STUDENT_SPONSOR
- 2.2 SEND_SPONSOR_LETTER
- 2.3 SEND_WELCOME_ABOARD_PACKAGE

- 3.1.1 RECORD_ARRIVAL
- 3.1.2 SETUP_STUDENT_COURSE_OF_STUDY
- 3.1.3 CHANGE_COURSE_IN_STUDENT_STUDY
- 3.1.4 CHANGE_REQT_OF_COURSE_REQUEST
- 3.1.5 RECORD_THESIS_PROPOSAL
- 3.1.6 REVISE_THESIS_PROPOSAL
- 3.1.7 ELIMINATE_JOINT_STATUS
- 3.1.8 REVISE_THESIS_TO_JOINT_STATUS
- 3.1.9 ARCHIVE_THESIS
- 3.1.10 REMOVE_ERRONEOUS_THESIS

- 3.2.1 ENTER DEPENDENT DATA
- 3.2.2 MODIFY DEPENDENT DATA
- 3.2.3 ELIMINATE DEPENDENT DATA
- 3.2.4 ENTER ACADEMIC BACKGROUND
- 3.2.5 MODIFY ACADEMIC BACKGROUND
- 3.2.6 REMOVE ACADEMIC BACKGROUND

- 3.2.7.1 RECORD MEDICAL INFO
- 3.2.7.2 RECORD LOCATION INFO
- 3.2.7.3 RECORD BIRTH INFO
- 3.2.7.4 RECORD STATION INFO
- 3.2.7.5 RECORD DESCRIPTIVE INFO
- 3.2.7.6 RECORD SECURITY INFO
- 3.2.7.7 UPDATE DEGREE INFO

- 3.3.1 RECORD NAVAL FITREP
- 3.3.2 RECORD NAVAL OFFICER DESCRIPTION
- 3.3.3 RECORD PRT
- 3.3.4 MODIFY PRT
- 3.3.5 REMOVE PRT
- 3.3.6 SETUP NAVAL BOOK REIMBURSEMENT
- 3.3.7 FILE BOOK CLAIM
- 3.3.8 MODIFY BOOK CLAIM
- 3.3.9 REMOVE BOOK CLAIM

- 4.1 SCHEDULE COURSE
- 4.2 POST GRADE
- 4.3 ESTABLISH A QUARTER QPR
- 4.4 REMOVE GRADUATES
- 4.5 MODIFY GRADE

- 5.1.1 ESTABLISH NEW CURRICULAR OFFICE
- 5.1.2 MODIFY CURRICULAR OFFICE
- 5.1.3 REMOVE CURRICULAR OFFICE
- 5.1.4 SETUP NEW CURRICULUM
- 5.1.5 MODIFY CURRICULUM
- 5.1.6 ELIMINATE CURRICULUM

- 5.2.1 SETUP TYPICAL COURSE OF STUDY
- 5.2.2 MODIFY TYPICAL COURSE OF STUDY
- 5.2.3 REMOVE TYPICAL COURSE OF STUDY

- 6.1 ESTABLISH A NEW COURSE
- 6.2 MODIFY EXISTING COURSE
- 6.3 REMOVE COURSE FROM CATALOG

- 7.1 MODIFY PASSWORD
- 7.2 ASSIGN NAVY BOOK CEILING
- 7.3 MODIFY NAVY BOOK CEILING

Process: ASSIGN_PROJECTED_STUDENT

This process involves entering a prospective student into the Naval Postgraduate Schools files.
(Primarily Admission's responsibility, however, Curricular Officers may add a student)

```
┌─ ASSIGN_PROJECTED_STUDENT
  │   IMPORTS: ...
  │   EXPORTS: ...
  │   LOCALS:
  │   ENTITY ACTIONS: ...
  │
  │   ┌─ READ curriculum
  │     │   WHERE DESIRED curriculum number IS EQUAL TO
  │     │       input_responsible curriculum number
  │   ┌─ WHEN successful
  │     │   MOVE curriculum TO output curriculum
  │   ┌─ CREATE student
  │     │   ASSOCIATE WITH curriculum WHICH belongs_to IT
  │     │   SET ssn TO input student ssn
  │     │   SET type_refresher TO input student type_refresher
  │     │   SET present_status TO "P"
  │     │   SET type_officer TO input student type_officer
  │     │   SET received_orders_to_attend TO input student
  │     │       received_orders_to_attend
  │     │   SET last_name TO input student last_name
  │     │   SET first_name TO input student first_name
  │     │   SET middle_initial TO input student middle_initial
  │     │   SET shortname TO input student shortname
  │     │   SET rank TO input student rank
  │     │   SET anticipated_graduation_date TO input student
  │     │       anticipated_graduation_date
  │     │   SET proposed_nps_degree TO input student proposed_nps_degree
  │     │   SET nps_major TO input student nps_major
  │     │   SET comment1 TO input student comment1
  │     │   SET apc TO input student apc
  │     │   SET commissioning_source TO input student
  │     │       commissioning_source
  │     │   SET dual_degree TO input student dual_degree
  │     │   SET program TO input student program
  │     │   SET international_service_component TO input student
  │     │       international_service_component
  │     │   SET country TO input student country
  │     │   SET service TO input student service
  │     │   SET convening_date TO input student convening_date
  │   ┌─ WHEN successful
  │     │   ┌─ IF student type_refresher IS EQUAL TO 1
  │     │       │   OR student type_refresher IS EQUAL TO 2
```

```

[ UPDATE student
  SET started_parent_curriculum TO "NA"
  SET completed_first_refresher_qtr TO "NA"
- ELSE IF student_type_refresher IS EQUAL TO 3
  [ UPDATE student
    SET completed_first_refresher_qtr TO "NA"
    SET started_parent_curriculum TO "N"
  - ELSE
    [ UPDATE student
      SET completed_first_refresher_qtr TO "N"
      SET started_parent_curriculum TO "N"
    - MOVE student TO output_student
      EXIT STATE IS successful_operation
    - WHEN already_exists
      EXIT STATE IS student_ae
    - WHEN not_found
      EXIT STATE IS curriculum_nf
  ]
]

```

Process: MODIFY_PROJECTED_STUDENT

This process modifies the description of a prospective student. (Admission or Curricular Officer)

```
MODIFY_PROJECTED_STUDENT
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

-- READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
-- WHEN successful
    UPDATE student
    SET type_refresher TO input student type_refresher
    SET type_officer TO input student type_officer
    SET received_orders_to_attend TO input student
        received_orders_to_attend
    SET last_name TO input student last_name
    SET first_name TO input student first_name
    SET middle_initial TO input student middle_initial
    SET shortname TO input student shortname
    SET rank TO input student rank
    SET anticipated_graduation_date TO input student
        anticipated_graduation_date
    SET nps_major TO input student nps_major
    SET proposed_nps_degree TO input student proposed_nps_degree
    SET comment1 TO input student comment1
    SET apc TO input student apc
    SET commissioning_source TO input student
        commissioning_source
    SET dual_degree TO input student dual_degree
    SET country TO input student country
    SET international_service_component TO input student
        international_service_component
    SET program TO input student program
    SET service TO input student service
    -- WHEN successful
    MOVE student TO output student
    EXIT STATE IS successful_operation
    -- WHEN not unique
    EXIT STATE IS student_nu
-- WHEN not found
    EXIT STATE IS student_nf
```

Process: REMOVE_PROJECTED_STUDENT

System Gen: This process removes an erroneously entered student. (Outside Scope: Admissions)

```
REMOVE PROJECTED_STUDENT
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

  READ student
    WHERE DESIRED student ssn IS EQUAL TO input_removing
      student ssn
  WHEN successful
    IF student present_status IS EQUAL TO "P"
      MOVE student TO removed student
      DELETE student
      EXIT STATE IS successful_operation
    ELSE
      EXIT STATE IS not_projected
  WHEN not found
    EXIT STATE IS student_nf
```

Process: ASSIGN_STUDENT_SPONSOR

Each incoming student may or may not be assigned a student sponsor who will assist them. Both the incoming students records will reflect who will act as the sponsor and the sponsor's files will reflect that he/she acted as a sponsor.

```
ASSIGN_STUDENT_SPONSOR
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

  READ student
    WHERE DESIRED student ssn IS EQUAL TO incoming student
    ssn
  WHEN successful
    READ existing_assigned student
      WHERE DESIRED existing_assigned student ssn
      IS EQUAL TO assigned student ssn
    WHEN successful
      UPDATE student
        SET name_of_sponsor TO existing_assigned student
        last_name
        SET date_that_a_sponsor_was_assigned TO CURRENT_DATE
      WHEN successful
        MOVE student TO output_incoming student
        UPDATE existing_assigned student
          SET in_bound_student_sponsor TO "Y"
        WHEN successful
          MOVE existing_assigned student TO output_assigned
          student
          EXIT STATE IS successful_operation
        WHEN not unique
          EXIT STATE IS student_nu
      WHEN not unique
        EXIT STATE IS student_nu
    WHEN not found
      EXIT STATE IS student_nf
  WHEN not found
    EXIT STATE IS student_nf
```


Process: SEND_SPONSOR_LETTER

An incoming student's files will reflect when the student sponsor sent his introductory letter to the new student.

```
SEND_SPONSOR_LETTER
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ student
    WHERE DESIRED student ssn IS EQUAL TO incoming student
      ssn
  WHEN successful
    UPDATE student
      SET date_sponsor_letter_sent TO incoming student
        date_sponsor_letter_sent
    WHEN successful
      MOVE student TO output_incoming student
      EXIT STATE IS successful_operation
    WHEN not unique
      EXIT STATE IS student_nu
  WHEN not found
    EXIT STATE IS student_nf
```

Process: SEND_WELCOME_ABOARD_PACKAGE

The incoming student's files will reflect when a welcome aboard package was sent to the new student.

```
SEND_WELCOME_ABOARD_PACKAGE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ student
    WHERE DESIRED student ssn IS EQUAL TO incoming student
      ssn
  WHEN successful
    UPDATE student
      SET date_welcome_package_sent TO incoming student
        date_welcome_package_sent
    WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    WHEN not unique
      EXIT STATE IS student_nu
  WHEN not found
    EXIT STATE IS student_nf
```

Process: RECORD_ARRIVAL

This process involves the recording of the arrival (at the curricular office) of an incoming student. If this student is in the Navy; the Student Book Money is created.

```

[ RECORD_ARRIVAL
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO arrived student ssn
  [ WHEN successful
    [ UPDATE student
      SET present_status TO "A"
      SET date_reported_aboard TO arrived student
        date_reported_aboard
    [ WHEN successful
      MOVE student TO output_arrived student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  ]
]
```

Process: SETUP_STUDENT_COURSE_OF_STUDY

This process involves the creation of a student's request for all courses he/she will need at NPS. The typical course of study for the student's curriculum will be used as a guide.

```
SETUP_STUDENT_COURSE_OF_STUDY
IMPORTS: ...
EXPORTS: ...
LOCALS:
    Work View    temp ief_supplied
                count
    Entity View temp student_course_of_study
                academic_year
                academic_quarter
ENTITY ACTIONS: ...

    READ student
        WHERE DESIRED student ssn IS EQUAL TO input student ssn
    WHEN successful
        SET temp student_course_of_study academic_year TO year(student
            convening_date)
            CASE OF month(student convening_date)
            CASE 1
                SET temp student_course_of_study academic_quarter TO "WIN"
            CASE 2
                SET temp student_course_of_study academic_quarter TO "WIN"
            CASE 3
                SET temp student_course_of_study academic_quarter TO "SPR"
            CASE 4
                SET temp student_course_of_study academic_quarter TO "SPR"
            CASE 5
                SET temp student_course_of_study academic_quarter TO "SPR"
            CASE 6
                SET temp student_course_of_study academic_quarter TO "SUM"
            CASE 7
                SET temp student_course_of_study academic_quarter TO "SUM"
            CASE 8
                SET temp student_course_of_study academic_quarter TO "SUM"
            CASE 9
                SET temp student_course_of_study academic_quarter TO "FALL"
                SET temp student_course_of_study academic_year TO
                    year(student convening_date) + 1
            CASE 10
                SET temp student_course_of_study academic_quarter TO "FALL"
                SET temp student_course_of_study academic_year TO
                    year(student convening_date) + 1
```

```

CASE 11
SET temp student_course_of_study academic_quarter TO "FALL"
SET temp student_course_of_study academic_year TO
    year(student convening_date) + 1
CASE 12
SET temp student_course_of_study academic_quarter TO "WIN"
OTHERWISE

SET temp ief_supplied count TO 1
MOVE student TO output student
READ curriculum
    WHERE DESIRED curriculum belongs_to CURRENT student
WHEN successful
    READ typical_course_of_study
        WHERE DESIRED typical_course_of_study
            recommended_for CURRENT curriculum
        AND DESIRED typical_course_of_study
            refresher_requirements IS EQUAL TO CURRENT
            student_type_refresher
        AND DESIRED typical_course_of_study type student
            IS EQUAL TO CURRENT student_type_officer
    WHEN successful
        REPEAT
            READ qtr_of_typical_study
                WHERE DESIRED qtr_of_typical_study
                    quarter_number IS EQUAL TO temp
                    ief_supplied count
                AND DESIRED qtr_of_typical_study makes_up
                    CURRENT typical_course_of_study
            WHEN successful
                READ EACH composition_of_typical_study
                    TARGETING outgroup_study
                    FROM THE BEGINNING UNTIL FULL
                WHERE DESIRED composition_of_typical_study
                    comprises CURRENT qtr_of_typical_study
            READ course
                WHERE DESIRED course reflected_in
                    CURRENT composition_of_typical_study
            WHEN successful
                MOVE course TO output course
                CREATE student_course_of_study
                ASSOCIATE WITH student WHICH possesses IT
                ASSOCIATE WITH course WHICH assigned_to IT
                SET status TO "R"
                SET academic_year TO temp
                    student_course_of_study
                    academic_year
                SET academic_quarter TO temp
                    student_course_of_study
                    academic_quarter

```

```

- WHEN successful
  MOVE student_course_of_study TO provided
    student_course_of_study
  EXIT STATE IS successful_operation
- WHEN already exists
  EXIT STATE IS student_course_of_study_ae
- WHEN not found
  EXIT STATE IS course_nf
-
  SET temp ief_supplied count TO (1 + temp
    ief_supplied count)
  IF temp student_course_of_study academic_quarter
    IS EQUAL TO "WIN"
    SET temp student_course_of_study
      academic_quarter TO "SPR"
  ELSE IF temp student_course_of_study
    academic_quarter IS EQUAL TO "SPR"
    SET temp student_course_of_study
      academic_quarter TO "SUM"
  ELSE IF temp student_course_of_study
    academic_quarter IS EQUAL TO "SUM"
    SET temp student_course_of_study
      academic_quarter TO "FALL"
  SET temp student_course_of_study academic_year
    TO (1 + temp student_course_of_study
      academic_year)
  ELSE
    SET temp student_course_of_study
      academic_quarter TO "WIN"
  - WHEN not found
    EXIT STATE IS stop
  - UNTIL EXITSTATE IS EQUAL TO stop
  - WHEN not found
    EXIT STATE IS typical_course_of_study_nf
  - WHEN not found
    EXIT STATE IS curriculum_nf
  - WHEN not found
    EXIT STATE IS student_nf

```

Process: CHANGE_COURSE_IN_STUDENT_STUDY

This process modifies the requested courses of a student.

CHANGE_COURSE_IN_STUDENT_STUDY

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ student_course_of_study

WHERE DESIRED student_course_of_study belongs_to
SOME student

AND THAT student ssn IS EQUAL TO input student ssn

AND DESIRED student_course_of_study composed_of
SOME course

AND THAT course number IS EQUAL TO present_input course
number

AND THAT course academic_department_code IS EQUAL TO
present_input course academic_department_code

WHEN successful

READ present course

WHERE DESIRED present course assigned_to
CURRENT student_course_of_study

WHEN successful

READ new course

WHERE DESIRED new course number IS EQUAL TO
new_input course number

AND DESIRED new course academic_department_code
IS EQUAL TO new_input course
academic_department_code

WHEN successful

TRANSFER student_course_of_study

FROM present course WHICH assigned_to IT
TO new course WHICH assigned_to IT

UPDATE student_course_of_study

SET validation TO new_input student_course_of_study
validation

SET pass_fail TO new_input student_course_of_study
pass_fail

SET academic_year TO new_input student_course_of_study
academic_year

SET academic_quarter TO new_input
student_course_of_study academic_quarter

WHEN successful

MOVE student_course_of_study TO output_new
student_course_of_study

EXIT STATE IS successful_operation

Process: CHANGE_REQT_OF_COURSE_REQUEST

This process will not request a different course, but will allow a change in requirements such as pass-fail, validation, academic quarter and year.

CHANGE_REQT_OF_COURSE_REQUEST

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ student_course_of_study

WHERE DESIRED student_course_of_study belongs_to
SOME student

AND THAT student ssn IS EQUAL TO designated student ssn

AND DESIRED student_course_of_study composed_of
SOME course

AND THAT course number IS EQUAL TO present_input course
number

AND THAT course academic_department_code IS EQUAL TO
present_input course academic_department_code

WHEN successful

UPDATE student_course_of_study

SET validation TO input_new student_course_of_study
validation

SET pass_fail TO input_new student_course_of_study pass_fail

SET academic_year TO input_new student_course_of_study
academic_year

SET academic_quarter TO input_new student_course_of_study
academic_quarter

WHEN successful

MOVE student_course_of_study TO output_new
student_course_of_study

EXIT STATE IS successful_operation

WHEN not unique

EXIT STATE IS student_course_of_study_nu

WHEN not found

EXIT STATE IS student_course_of_study_nf

Process: RECORD_THESIS_PROPOSAL

System Gen: This process records the creation of a thesis for a particular student or students. (Outside Scope: Thesis Processor)

```

[ RECORD_THESIS_PROPOSAL
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO identified student
      ssn
  ]
  [ WHEN successful
    MOVE student TO export student
    [ CREATE thesis
      ASSOCIATE WITH student WHICH writes IT
      SET number USING thesis_number
        WHICH IMPORTS: Entity View indicated thesis
      SET due_date_year TO indicated thesis due_date_year
      SET due_date_month TO indicated thesis due_date_month
      SET title TO indicated thesis title
      SET status TO indicated thesis status
      SET advisor TO indicated thesis advisor
      SET second_reader TO indicated thesis second_reader
      SET classified TO indicated thesis classified
      SET joint TO indicated thesis joint
    ]
    [ WHEN successful
      MOVE thesis TO output thesis
      EXIT STATE IS successful_operation
    ]
    [ WHEN already exists
      EXIT STATE IS thesis_ae
    ]
  ]
  [ WHEN not found
    EXIT STATE IS student_nf
  ]
]
```

Process: REVISE_THESIS_PROPOSAL

System Gen: This process modifies an existing thesis.
If a student must be removed from a joint thesis, a
disassociation would be required. (Outside scope:
Thesis Processor)

```

[ REVISE_THESIS_PROPOSAL
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

    [ READ thesis
      WHERE DESIRED thesis due_date_year IS EQUAL TO input
        thesis due_date_year
      AND DESIRED thesis number IS EQUAL TO input thesis number

    [ WHEN successful
      [ UPDATE thesis
        SET title TO input thesis title
        SET status TO input thesis status
        SET advisor TO input thesis advisor
        SET second_reader TO input thesis second_reader
        SET classified TO input thesis classified
      [ WHEN successful
        MOVE thesis TO output thesis
        EXIT STATE IS successful_operation
      [ WHEN not unique
        EXIT STATE IS thesis_nu
      ]
    ]

    [ WHEN not found
      EXIT STATE IS thesis_nf
    ]
  ]

```

Process: ELIMINATE_JOINT_STATUS

System Gen: This process involves the disassociation of one student from a joint thesis. (Outside scope: Thesis Processor)

```
ELIMINATE_JOINT_STATUS
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ thesis
    WHERE DESIRED thesis due_date_year IS EQUAL TO input
      thesis due_date_year
    AND DESIRED thesis number IS EQUAL TO input thesis number
    AND DESIRED thesis joint IS EQUAL TO "Y"
  WHEN successful
    READ existing student
      WHERE DESIRED existing student ssn IS EQUAL TO
        input_removing student ssn
    WHEN successful
      DISASSOCIATE thesis
        FROM existing student WHICH writes IT
      UPDATE thesis
        SET joint TO "N"
      WHEN successful
        MOVE thesis TO output thesis
        EXIT STATE IS successful_operation
      WHEN not unique
        EXIT STATE IS thesis_nu
    WHEN not found
      EXIT STATE IS student_nf
  WHEN not found
    EXIT STATE IS thesis_nf
```

Process: REVISE_THESIS_TO_JOINT_STATUS

System Gen: This process revises a thesis entity to reflect a joint status and associates the thesis with an additional student. (Outside scope: Thesis Processor)

```

[ REVISE_THESIS_TO_JOINT_STATUS
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

    [ READ thesis
      WHERE DESIRED thesis due_date_year IS EQUAL TO input
        thesis due_date_year
      AND DESIRED thesis number IS EQUAL TO input thesis number
    [ WHEN successful
      [ READ student
        WHERE DESIRED student ssn IS EQUAL TO input_additional
          student ssn
      [ WHEN successful
        ASSOCIATE student
          WITH thesis WHICH written_by IT
        [ UPDATE thesis
          SET joint TO "Y"
        [ WHEN successful
          MOVE thesis TO output thesis
        [ WHEN not unique
          EXIT STATE IS thesis_nu
        MOVE student TO output_additional student
        EXIT STATE IS successful_operation
      [ WHEN not found
        EXIT STATE IS student_nf
    [ WHEN not found
      EXIT STATE IS thesis_nf
  ]
]
```

Process: ARCHIVE_THESIS

System Gen: This process archives thesis listing for those students who have been archived. (Outside scope: Thesis Processor)

```

[ ARCHIVE_THESIS
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ thesis
    WHERE DESIRED thesis due_date_year IS EQUAL TO input
      thesis due_date_year
    AND DESIRED thesis number IS EQUAL TO input thesis number
  ]
  [ WHEN successful
    MOVE thesis TO output_removed thesis
    DELETE thesis
    EXIT STATE IS successful_operation
  ]
  [ WHEN not found
    EXIT STATE IS thesis_nf
  ]
]
```


Process: REMOVE_ERRONEOUS_THESIS

System Gen: This process removes a thesis which has been erroneously entered or has been abandoned by its author, as distinguished from a mere modification.
(Outside scope: Thesis Processor)

REMOVE_ERRONEOUS_THESIS

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ thesis

WHERE DESIRED thesis due_date_year IS EQUAL TO input
thesis due_date_year

AND DESIRED thesis number IS EQUAL TO input thesis number

WHEN successful

MOVE thesis TO output thesis

DELETE thesis

EXIT STATE IS successful_operation

WHEN not found

EXIT STATE IS thesis_nf

Process: ENTER_DEPENDENT_DATA

This process involves the creation of student's dependent information.

```
ENTER_DEPENDENT_DATA
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

-- READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
-- WHEN successful
    MOVE student TO output student
    [
    CREATE dependent
    ASSOCIATE WITH student WHICH possesses IT
    SET last_name TO input dependent last_name
    SET first_name TO input dependent first_name
    SET family_member TO input dependent family_member
    -- WHEN successful
    MOVE dependent TO output dependent
    EXIT STATE IS successful_operation
    -- WHEN already exists
    EXIT STATE IS dependent_ae
    ]
-- WHEN not found
    EXIT STATE IS student_nf
```

Process: MODIFY_DEPENDENT_DATA

This process involves the modification of a student's dependent information.

```
-- MODIFY_DEPENDENT_DATA
```

```
IMPORTS: ...
```

```
EXPORTS: ...
```

```
LOCALS:
```

```
ENTITY ACTIONS: ...
```

```
-- READ dependent
```

```
WHERE DESIRED dependent last_name IS EQUAL TO input  
dependent last_name
```

```
AND DESIRED dependent first_name IS EQUAL TO input  
dependent first_name
```

```
AND DESIRED dependent_belongs_to SOME student
```

```
AND THAT student ssn IS EQUAL TO input student ssn
```

```
-- WHEN successful
```

```
-- UPDATE dependent
```

```
SET family_member TO input dependent family_member
```

```
-- WHEN successful
```

```
MOVE dependent TO output dependent
```

```
EXIT STATE IS successful_operation
```

```
-- WHEN not unique
```

```
EXIT STATE IS dependent_nu
```

```
-- WHEN not found
```

```
EXIT STATE IS dependent_nf
```

Process: ELIMINATE_DEPENDENT_DATA

This process involves the removal of an entity which is no longer a dependent of a student.

```
ELIMINATE_DEPENDENT_DATA
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

  READ dependent
    WHERE DESIRED dependent last_name IS EQUAL TO input
      dependent last_name
    AND DESIRED dependent first_name IS EQUAL TO input
      dependent first_name
    AND DESIRED dependent belongs_to SOME existing student
    AND THAT existing student ssn IS EQUAL TO input student
      ssn
  WHEN successful
    MOVE dependent TO output_removed dependent
    DELETE dependent
    EXIT STATE IS successful_operation
  WHEN not found
    EXIT STATE IS dependent_nf
```

Process: ENTER_ACADEMIC_BACKGROUND

This process involves the recording of a student's prior academic history.

```
ENTER_ACADEMIC_BACKGROUND
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  WHEN successful
    MOVE student TO output student
    CREATE academic_history
    ASSOCIATE WITH student WHICH possesses IT
    SET school TO input academic_history school
    SET degree TO input academic_history degree
    SET major TO input academic_history major
    SET gpa TO input academic_history gpa
    SET date TO input academic_history date
  WHEN successful
    MOVE academic_history TO output academic_history
    EXIT STATE IS successful_operation
  WHEN already exists
    EXIT STATE IS academic_history_ae
  WHEN not found
    EXIT STATE IS student_nf
```

Process: MODIFY_ACADEMIC_BACKGROUND

This process involves the modification of a student's record of their academic history.

```

[ MODIFY_ACADEMIC_BACKGROUND
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

    [ READ academic_history
      WHERE DESIRED academic_history major IS EQUAL TO
        present_input academic_history major
      AND DESIRED academic_history degree IS EQUAL TO
        present_input academic_history degree
      AND DESIRED academic_history_belongs_to SOME student
      AND THAT student ssn IS EQUAL TO input student ssn

    [ WHEN successful
      [ UPDATE academic_history
        SET school TO input_adjusting academic_history school
        SET gpa TO input_adjusting academic_history gpa
        SET date TO input_adjusting academic_history date
      [ WHEN successful
        MOVE academic_history TO output academic_history
        EXIT STATE IS successful_operation
      [ WHEN not unique
        EXIT STATE IS academic_history_nu

    [ WHEN not found
      EXIT STATE IS academic_history_nf

```

Process: REMOVE_ACADEMIC_BACKGROUND

This process removes an erroneously entered student academic history.

```
REMOVE_ACADEMIC_BACKGROUND
```

```
IMPORTS: ...
```

```
EXPORTS: ...
```

```
LOCALS:
```

```
ENTITY ACTIONS: ...
```

```
  READ academic_history
```

```
    WHERE DESIRED academic_history major IS EQUAL TO
```

```
      input_removing academic_history major
```

```
    AND DESIRED academic_history degree IS EQUAL TO
```

```
      input_removing academic_history degree
```

```
    AND DESIRED academic_history belongs_to SOME student
```

```
    AND THAT student ssn IS EQUAL TO input student ssn
```

```
  WHEN successful
```

```
    MOVE academic_history TO output_removed academic_history
```

```
    DELETE academic_history
```

```
    EXIT STATE IS successful_operation
```

```
  WHEN not found
```

```
    EXIT STATE IS academic_history_nf
```


Process: RECORD_MEDICAL_INFO

This process updates the medical information maintained on a particular student.

```

[ RECORD_MEDICAL_INFO
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE student
      SET aids_test_date TO input student aids_test_date
      SET physical_date TO input student physical_date
      SET dental_date TO input student dental_date
    [ WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  ]
]
```

Process: RECORD_LOCATION_INFO

This process records or updates the local address, phone number, section number, study space, or locker number of a particular student.

```

[ RECORD_LOCATION_INFO
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE student
      SET phone_number TO input student phone_number
      SET street TO input student street
      SET city TO input student city
      SET zip_code TO input student zip_code
      SET lamesa_housing_occupant TO input student
        lamesa_housing_occupant
      SET smc_box_number TO input student smc_box_number
      SET section_number TO input student section_number
      SET split_section TO input student split_section
      SET study_space TO input student study_space
      SET locker_number TO input student locker_number
    [ WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  ]
]
```

Process: RECORD_BIRTH_INFO

This process records or updates the date of birth and place of birth of a particular student.

```

[ RECORD_BIRTH_INFO
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE student
      SET date_of_birth TO input student date_of_birth
      SET place_of_birth_city TO input student place_of_birth_city
      SET place_of_birth_state TO input student
        place_of_birth_state
    [ WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  ]
]
```

Process: RECORD_STATION_INFO

This process records the previous duty station, next duty station and date of orders of a particular student.

```

[ RECORD_STATION_INFO
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE student
      SET previous_duty_station TO input student
        previous_duty_station
      SET date_of_orders TO input student date_of_orders
      SET next_duty_station TO input student next_duty_station
    [ WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  ]
]
```

Process: RECORD_DESCRIPTIVE_INFO

This process updates descriptive information of a particular student. (ex. name, gender, rank, commissioning source, etc.)

```
RECORD_DESCRIPTIVE_INFO
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  -- READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  -- WHEN successful
    UPDATE student
    SET type_refresher TO input student type_refresher
    SET present_status TO input student present_status
    SET type_officer TO input student type_officer
    SET received_orders_to_attend TO input student
      received_orders_to_attend
    SET last_name TO input student last_name
    SET first_name TO input student first_name
    SET middle_initial TO input student middle_initial
    SET shortname TO input student shortname
    SET gender TO input student gender
    SET rank TO input student rank
    SET date_of_rank TO input student date_of_rank
    SET marital_status TO input student marital_status
    SET commissioning_source TO input student
      commissioning_source
    SET library_card_number TO input student library_card_number
    SET mainframe_account_number TO input student
      mainframe_account_number
    -- WHEN successful
    MOVE student TO output student
    EXIT STATE IS successful_operation
    -- WHEN not unique
    EXIT STATE IS student_nu
  -- WHEN not found
  EXIT STATE IS student_nf
```

Process: RECORD_SECURITY_INFO

This process records the security attributes of a particular student. (ex. background, access)

RECORD_SECURITY_INFO

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ student

WHERE DESIRED student ssn IS EQUAL TO input student ssn

WHEN successful

UPDATE student

SET security_background TO input student security_background

SET security_access TO input student security_access

WHEN successful

MOVE student TO output student

EXIT STATE IS successful_operation

WHEN not unique

EXIT STATE IS student_nu

WHEN not found

EXIT STATE IS student_nf

Process: UPDATE_DEGREE_INFO

This process updates a student's major, type degree, accreditation, or dual degree status.

```

[ UPDATE_DEGREE_INFO
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...
    [ READ student
      WHERE DESIRED student ssn IS EQUAL TO input student ssn
    ]
    [ WHEN successful
      [ UPDATE student
        SET proposed_nps_degree TO input student proposed_nps_degree
        SET accreditation_status TO input student
          accreditation_status
        SET nps_major TO input student nps_major
        SET dual_degree TO input student dual_degree
      ]
      [ WHEN successful
        MOVE student TO output student
      ]
      [ EXIT STATE IS successful_operation
      ]
      [ WHEN not unique
        EXIT STATE IS student_nu
      ]
    ]
    [ WHEN not found
      EXIT STATE IS student_nf
    ]
  ]

```


Process: RECORD_NAVAL_FITREP

This process records the date that a Naval Fitness Report was submitted, and the due date of the next report.

```

[ RECORD_NAVAL_FITREP
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE student
      SET last_fitrep_date TO input student last_fitrep_date
      SET next_fitrep_due TO input student next_fitrep_due
    [ WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  [
```

Process: RECORD_NAVAL_OFFICER_DESCRIPTION

This process involves the recording of a Naval Officer's Lineal-Number, Year-Group, and Officer-Designator.

```

[ RECORD_NAVAL_OFFICER_DESCRIPTION
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE student
      SET officer_designator TO input student officer_designator
      SET lineal_number TO input student lineal_number
      SET officer_year_group TO input student officer_year_group
    [ WHEN successful
      MOVE student TO output student
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS student_nu
    [ WHEN not found
      EXIT STATE IS student_nf
  ]
]
```

Process: RECORD_PRT

This process involves the creation of a Naval Student's Physical Readiness Training Record.

```

[ RECORD_PRT
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  [ WHEN successful
    MOVE student TO output student
    [ CREATE prt
      ASSOCIATE WITH student WHICH takes IT
      SET score TO input prt score
      SET body_fat TO input prt body_fat
      SET date_of_test TO input prt date_of_test
    [ WHEN successful
      MOVE prt TO output prt
      EXIT STATE IS successful_operation
    [ WHEN already exists
      EXIT STATE IS prt_ae
  [ WHEN not found
    EXIT STATE IS student_nf

```

Process: MODIFY_PRT

This process modifies a Naval Student's Physical Readiness Training record.

```

[ MODIFY_PRT
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ prt
    WHERE DESIRED prt date_of_test IS EQUAL TO input prt
      date_of_test
    AND DESIRED prt taken_by SOME navy
    AND THAT navy ssn IS EQUAL TO input student ssn
  [ WHEN successful
    [ UPDATE prt
      SET body_fat TO input prt body_fat
      SET score TO input prt score
    [ WHEN successful
      MOVE prt TO output prt
      EXIT STATE IS successful_operation
    [ WHEN not unique
      EXIT STATE IS prt_nu
  [ WHEN not found
    EXIT STATE IS prt_nf

```

Process: REMOVE_PRT

This process removes an erroneously entered PRT.

```
REMOVE_PRT
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

  READ prt
    WHERE DESIRED prt date_of_test IS EQUAL TO input prt
      date_of_test
    AND DESIRED prt taken_by SOME navy
    AND THAT navy ssn IS EQUAL TO input student ssn
  WHEN successful
    MOVE prt TO output prt
    DELETE prt
    EXIT STATE IS successful_operation
  WHEN not found
    EXIT STATE IS prt_nf
```

Process: SETUP_NAVAL_BOOK_REIMBURSEMENT

This process creates a student's allocated book money for an academic year. (prorated based on the number of quarters remaining in the academic year or on the time a student entered in the academic year).

```
SETUP_NAVAL_BOOK_REIMBURSEMENT
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ student
    WHERE DESIRED student ssn IS EQUAL TO input student ssn
  WHEN successful
    READ total_annual_naval_book_ceiling
      WHERE DESIRED total_annual_naval_book_ceiling
        date_implemented IS EQUAL TO input
        total_annual_naval_book_ceiling date_implemented
    WHEN successful
      CREATE student_book_reimbursement
      ASSOCIATE WITH total_annual_naval_book_ceiling
        WHICH sets_limit IT
      ASSOCIATE WITH student WHICH obtains IT
      SET year TO input student_book_reimbursement year
      SET total_amount_eligible TO
        ((total_annual_naval_book_ceiling
          total_amount / 4) *
          student_book_reimbursement
          number_academic_qtrs_authorized)
      WHEN successful
        MOVE student_book_reimbursement TO output
          student_book_reimbursement
      EXIT STATE IS successful_operation
      WHEN already exists
        EXIT STATE IS student_book_money_ae
    WHEN not found
      EXIT STATE IS navy_book_eligibility_nf
  WHEN not found
    EXIT STATE IS student_nf
```

Process: FILE_BOOK_CLAIM

This process creates a claim against a Naval student's book money (total amt they are allowed to spend for an academic year) and reduces the amount remaining in the student's book money.

FILE_BOOK_CLAIM

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ student_book_reimbursement

WHERE DESIRED student_book_reimbursement year IS EQUAL TO
input student_book_reimbursement year

AND DESIRED student_book_reimbursement provided_to
SOME navy

AND THAT navy ssn IS EQUAL TO input student ssn

WHEN successful

IF input book_claim amount_of_claim IS LESS OR EQUAL TO
student_book_reimbursement amount_remaining

CREATE book_claim

ASSOCIATE WITH student_book_reimbursement

WHICH depreciated by IT

SET amount_of_claim TO input book_claim amount_of_claim

SET academic_quarter TO input book_claim academic_quarter

WHEN successful

MOVE book_claim TO output book_claim

EXIT STATE IS successful_operation

WHEN already exists

EXIT STATE IS book_claim_ae

ELSE

EXIT STATE IS book_claim_too_large

WHEN not found

EXIT STATE IS student_book_money_nf

Process: MODIFY_BOOK_CLAIM

This process modifies an existing Naval student's book claim and makes the needed adjustment in the student's book money.

```

-- MODIFY_BOOK_CLAIM
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  -- READ book_claim
    WHERE DESIRED book_claim academic_quarter IS EQUAL TO
      input_new book_claim academic_quarter
    AND DESIRED book_claim reduces
      SOME student_book_reimbursement
    AND THAT student_book_reimbursement year IS EQUAL TO
      input student_book_reimbursement year
    AND THAT student_book_reimbursement provided_to SOME navy
    AND THAT navy ssn IS EQUAL TO input student ssn

  -- WHEN successful
    IF student_book_reimbursement amount_remaining
      IS GREATER OR EQUAL TO input_new book_claim
      amount_of_claim - book_claim amount_of_claim
      OR input_new book_claim amount_of_claim IS LESS THAN
      book_claim amount_of_claim
      -- UPDATE book_claim
        SET amount_of_claim TO input_new book_claim
        amount_of_claim
      -- WHEN successful
        MOVE book_claim TO output book_claim
        EXIT STATE IS successful_operation
      -- WHEN not unique
        EXIT STATE IS book_claim_nu
    ELSE
      EXIT STATE IS book_claim_too_large

  -- WHEN not found
    EXIT STATE IS book_claim_nf

```


Process: REMOVE_BOOK_CLAIM

This process removes an erroneously entered book claim.

REMOVE_BOOK_CLAIM

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ book_claim

WHERE DESIRED book_claim academic_quarter IS EQUAL TO
input book_claim academic_quarter

AND DESIRED book_claim reduces
SOME student_book_reimbursement

AND THAT student_book_reimbursement year IS EQUAL TO
input student_book_reimbursement year

AND THAT student_book_reimbursement provided_to SOME navy
AND THAT navy ssn IS EQUAL TO input student ssn

WHEN successful

MOVE book_claim TO output book_claim

DELETE book_claim

EXIT STATE IS successful_operation

WHEN not found

EXIT STATE IS book_claim_nf

Process: SCHEDULE_COURSE

System Gen: This process involves scheduling a course requested by a specific student. (Outside scope: Registrar)

```

[ SCHEDULE_COURSE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student_course_of_study
    WHERE DESIRED student_course_of_study belongs_to
      SOME student
    AND THAT student ssn IS EQUAL TO input student ssn
    AND DESIRED student_course_of_study composed_of
      SOME course
    AND THAT course number IS EQUAL TO input course number
    AND THAT course academic_department_code IS EQUAL TO
      input course academic_department_code
    [ WHEN successful
      [ UPDATE student_course_of_study
        SET status TO "S"
        SET section_number TO input student_course_of_study
          section_number
        [ WHEN successful
          MOVE student_course_of_study TO output
            student_course_of_study
          EXIT STATE IS successful_operation
        [ WHEN not unique
          EXIT STATE IS student_course_of_study_nu
        [ WHEN not found
          EXIT STATE IS student_course_of_study_nf
      ]
    ]
  ]
]
```

Process: POST_GRADE

System Gen: This process records the grade a student earned at the completion of a course. (Outside scope: Registrar)

```
POST_GRADE
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

-- READ student_course_of_study
    WHERE DESIRED student_course_of_study belongs_to
        SOME student
    AND THAT student_ssn IS EQUAL TO input student_ssn
    AND DESIRED student_course_of_study composed_of
        SOME course
    AND THAT course number IS EQUAL TO input course number
    AND THAT course_academic_department_code IS EQUAL TO
        input course_academic_department_code
-- WHEN successful
    UPDATE student_course_of_study
    SET status TO "C"
    SET grade TO input student_course_of_study grade
    -- WHEN successful
    MOVE student_course_of_study TO output
        student_course_of_study
    EXIT STATE IS successful_operation
    -- WHEN not unique
    EXIT STATE IS student_course_of_study_nu
-- WHEN not found
    EXIT STATE IS student_course_of_study_nf
```

Process: ESTABLISH_A_QUARTER_QPR

System Gen: This process involves the creation of an QPR for a particular quarter and year. This process would be called when no QPR exist for that particular quarter and year when a grade is posted. (Outside scope: Registrar)

```
ESTABLISH_A_QUARTER_QPR
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  -- READ student
    WHERE DESIRED student ssn IS EQUAL TO import student ssn
  -- WHEN successful
    READ student_course_of_study
      WHERE DESIRED student_course_of_study belongs_to
        CURRENT student
      AND DESIRED student_course_of_study composed_of
        SOME course
      AND THAT course number IS EQUAL TO input course number
      AND THAT course academic_department_code IS EQUAL TO
        input course academic_department_code
    -- WHEN successful
      CREATE quarter_qpr
      ASSOCIATE WITH student_course_of_study
        WHICH used_to_calculate IT
      ASSOCIATE WITH student WHICH earns IT
      SET academic_year TO input quarter_qpr academic_year
      SET academic_quarter TO input quarter_qpr
        academic_quarter
      -- WHEN successful
        MOVE quarter_qpr TO export quarter_qpr
        EXIT STATE IS successful_operation
      -- WHEN already exists
        EXIT STATE IS quarter_qpr_ae
    -- WHEN not found
      EXIT STATE IS student_course_of_study_nf
  -- WHEN not found
    EXIT STATE IS student_nf
```

Process: REMOVE_GRADUATES

System Gen: This process archives the records of those students who have graduated (or those who attended but did not meet the requirements for graduation) from the Naval Postgraduate School. (Outside scope: Registrar)

Procedure actually accomplished by Registrar ONLY, however, must be modeled here to enable view capability by curricular officers.

REMOVE_GRADUATES

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

-- READ student

WHERE DESIRED student ssn IS EQUAL TO input student ssn

-- WHEN successful

MOVE student TO output student

DELETE student

EXIT STATE IS successful_operation

-- WHEN not found

EXIT STATE IS student_nf

Process: MODIFY_GRADE

System Gen: This process is a generic update of a student's grade. (Outside scope: Registrar)

```

[ MODIFY_GRADE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ student_course_of_study
    WHERE DESIRED student_course_of_study belongs_to
      SOME student
    AND THAT student ssn IS EQUAL TO input student ssn
    AND DESIRED student_course_of_study composed_of
      SOME course
    AND THAT course number IS EQUAL TO input course number
    AND THAT course academic_department_code IS EQUAL TO
      input course academic_department_code

    [ WHEN successful
      [ UPDATE student_course_of_study
        SET grade TO input student_course_of_study grade
        [ WHEN successful
          MOVE student_course_of_study TO output
            student_course_of_study
          EXIT STATE IS successful_operation
        [ WHEN not unique
          EXIT STATE IS student_course_of_study_nu
        ]
      ]
    ]

  [ WHEN not found
    EXIT STATE IS student_course_of_study_nf
  ]
]
```

Process: ESTABLISH_NEW_CURRICULAR_OFFICE

System Gen: This process creates a new curricular office in addition to the 11 present offices. Additionally, at least one curriculum must be created that composes that particular curricular office. This process requires a modification to the code to add a permitted value for a curricular office code. (Outside scope: Registrar with the assistance of the Curricular Officers and Academic Associates)

```
ESTABLISH_NEW_CURRICULAR_OFFICE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  - CREATE curricular_office
    SET title TO input curricular_office title
    SET code TO input curricular_office code
    SET password TO input curricular_office password
  - WHEN successful
    MOVE curricular_office TO output curricular_office
    FOR EACH group_import
      TARGETING group_export
        - CREATE curriculum
          ASSOCIATE WITH curricular_office WHICH composed_of IT
          SET title TO input curriculum title
          SET number TO input curriculum number
        - WHEN successful
          MOVE curriculum TO output curriculum
          EXIT STATE IS successful_operation
        - WHEN already exists
          EXIT STATE IS curriculum_ae
    - WHEN already exists
      EXIT STATE IS curricular_office_ae
```

Process: MODIFY_CURRICULAR_OFFICE

System Gen: This process modifies an existing
curricular office. (Outside scope: Registrar)

```
MODIFY_CURRICULAR_OFFICE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ curricular_office
    WHERE DESIRED curricular_office code IS EQUAL TO input
      curricular_office code
  WHEN successful
    UPDATE curricular_office
    SET title TO input curricular_office title
    WHEN successful
    MOVE curricular_office TO output curricular_office
    EXIT STATE IS successful_operation
    WHEN not unique
    EXIT STATE IS curricular_office_nu
  WHEN not found
    EXIT STATE IS curricular_office_nf
```


Process: REMOVE_CURRICULAR_OFFICE

System Gen: This process removes a curricular office.
(Outside scope: Registrar)

```
[ REMOVE_CURRICULAR_OFFICE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ curricular_office
    WHERE DESIRED curricular_office code IS EQUAL TO input
      curricular_office code
  [ WHEN successful
    MOVE curricular_office TO output curricular_office
    DELETE curricular_office
    EXIT STATE IS successful_operation
  [ WHEN not found
    EXIT STATE IS curricular_office_nf
  [
```

Process: SETUP_NEW_CURRICULUM

System Gen: This process involves the creation of a new curriculum for a particular curricular office.
(Outside scope: Registrar)

```

[ SETUP_NEW_CURRICULUM
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ curricular_office
    WHERE DESIRED curricular_office code IS EQUAL TO input
      curricular_office code
  ]
  [ WHEN successful
    MOVE curricular_office TO output curricular_office
    [ CREATE curriculum
      ASSOCIATE WITH curricular_office WHICH composed_of IT
      SET title TO input curriculum title
      SET number TO input curriculum number
    ]
    [ WHEN successful
      MOVE curriculum TO output curriculum
      EXIT STATE IS successful_operation
    ]
    [ WHEN already exists
      EXIT STATE IS curriculum_ae
    ]
  ]
  [ WHEN not found
    EXIT STATE IS curricular_office_nf
  ]
]
```

Process: MODIFY_CURRICULUM

System Gen: This process involves the modification of a curriculum for a particular curricular office.
(Outside scope: Registrar)

```

[ MODIFY_CURRICULUM
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ curriculum
    WHERE DESIRED curriculum number IS EQUAL TO input
      curriculum number
    [ WHEN successful
      [ UPDATE curriculum
        SET title TO input curriculum title
      [ WHEN successful
        MOVE curriculum TO output curriculum
        EXIT STATE IS successful_operation
      [ WHEN not unique
        EXIT STATE IS curriculum_nu
    [ WHEN not found
      EXIT STATE IS curriculum_nf
  ]
]
```

Process: ELIMINATE_CURRICULUM

System Gen: This process involves the removal of a curriculum from a particular curricular office.
(Outside scope: Registrar)

```

[ ELIMINATE_CURRICULUM
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ READ curriculum
    WHERE DESIRED curriculum number IS EQUAL TO input
      curriculum number
  ]
  [ WHEN successful
    MOVE curriculum TO output curriculum
    DELETE curriculum
    EXIT STATE IS successful_operation
  ]
  [ WHEN not found
    EXIT STATE IS curriculum_nf
  ]
]
```

Process: SETUP_TYPICAL_COURSE_OF_STUDY

This process involves the creation of a catalog identification of a new typical course of study for a particular curriculum.

```
SETUP_TYPICAL_COURSE_OF_STUDY
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  READ curriculum
    WHERE DESIRED curriculum number IS EQUAL TO input
      curriculum number
  WHEN successful
    MOVE curriculum TO output curriculum
    CREATE typical_course_of_study
    ASSOCIATE WITH curriculum WHICH recommends IT
    SET type_student TO input typical_course_of_study
      type_student
    SET refresher_requirements TO input typical_course_of_study
      refresher_requirements
  WHEN successful
    MOVE typical_course_of_study TO output
      typical_course_of_study
    FOR EACH group_import
      TARGETING group_export
      CREATE qtr_of_typical_study
      ASSOCIATE WITH typical_course_of_study
        WHICH identifies IT
      SET quarter_number TO input qtr_of_typical_study
        quarter_number
      WHEN successful
        MOVE qtr_of_typical_study TO output
          qtr_of_typical_study
      FOR EACH group_import_2
        TARGETING group_export_2
        READ course
          WHERE DESIRED course number IS EQUAL TO
            input course number
          AND DESIRED course academic_department_code
            IS EQUAL TO input course
              academic_department_code
        WHEN successful
          MOVE course TO output course
```


Process: MODIFY_TYPICAL_COURSE_OF_STUDY

This process involves the modification of a catalogued typical course of study for a particular curriculum.

MODIFY_TYPICAL_COURSE_OF_STUDY

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ composition_of_typical_study

WHERE DESIRED composition_of_typical_study comprises
SOME qtr_of_typical_study

AND THAT qtr_of_typical_study quarter_number IS EQUAL TO
import qtr_of_typical_study quarter_number

AND THAT qtr_of_typical_study makes_up
SOME typical_course_of_study

AND THAT typical_course_of_study refresher_requirements
IS EQUAL TO import typical_course_of_study
refresher_requirements

AND THAT typical_course_of_study type_student IS EQUAL TO
import typical_course_of_study type_student

AND THAT typical_course_of_study recommended_for
SOME curriculum

AND THAT curriculum number IS EQUAL TO import curriculum
number

AND DESIRED composition_of_typical_study identifies
SOME course

AND THAT course number IS EQUAL TO import course number

AND THAT course academic_department_code IS EQUAL TO
import course academic_department_code

WHEN successful

UPDATE composition_of_typical_study

SET type_of_course TO import composition_of_typical_study
type_of_course

WHEN successful

MOVE composition_of_typical_study TO export
composition_of_typical_study

READ persistent_2 course

WHERE DESIRED persistent_2 course reflected_in
CURRENT composition_of_typical_study

WHEN successful

MOVE persistent_2 course TO export course

READ persistent_3 course

WHERE DESIRED persistent_3 course number
IS EQUAL TO import_2 course number

AND DESIRED persistent_3 course

academic_department_code IS EQUAL TO

import_2_course academic_department_code

```

- WHEN successful
  TRANSFER composition_of_typical_study
    FROM persistent_2 course WHICH reflected_in IT
    TO persistent_3 course WHICH reflected_in IT
  MOVE persistent_3 course TO export_2 course
  EXIT STATE IS successful_operation
- WHEN not found
  EXIT STATE IS course_nf

- WHEN not found
  EXIT STATE IS course_nf

- WHEN not unique
  EXIT STATE IS composition_of_typical_study_nu

- WHEN not found
  EXIT STATE IS composition_of_typical_study_nf

```


Process: REMOVE_TYPICAL_COURSE_OF_STUDY

This process removes an erroneously entered typical course of study.

```
REMOVE_TYPICAL_COURSE_OF_STUDY
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

-- READ typical_course_of_study
   WHERE DESIRED typical_course_of_study
         refresher_requirements IS EQUAL TO input
         typical_course_of_study refresher_requirements
   AND DESIRED typical_course_of_study type_student
         IS EQUAL TO input typical_course_of_study
         type_student
   AND DESIRED typical_course_of_study recommended_for
         SOME curriculum
   AND THAT curriculum number IS EQUAL TO input curriculum
         number
-- WHEN successful
   MOVE typical_course_of_study TO output typical_course_of_study
   DELETE typical_course_of_study
   EXIT STATE IS successful_operation
-- WHEN not found
   EXIT STATE IS typical_course_of_study_nf
```

Process: ESTABLISH_A_NEW_COURSE

System Gen: This process creates a course. (Outside
scope: Registrar)

```
ESTABLISH_A_NEW_COURSE
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

  CREATE course
  SET name TO input course name
  SET lecture_credit_hours TO input course lecture_credit_hours
  SET lab_credit_hours TO input course lab_credit_hours
  SET academic_department_code TO input course
    academic_department_code
  SET number TO input course number
  WHEN successful
  MOVE course TO output course
  EXIT STATE IS successful_operation
  WHEN already exists
  EXIT STATE IS course_ae
```

Process: MODIFY_EXISTING_COURSE

System Gen: This process modifies an existing course.
(Outside scope: Registrar)

```
MODIFY_EXISTING_COURSE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  -- READ course
      WHERE DESIRED course number IS EQUAL TO input course
          number
      AND DESIRED course academic_department_code IS EQUAL TO
          input course academic_department_code
  -- WHEN successful
      UPDATE course
      SET name TO input course name
      SET lecture_credit_hours TO input course
          lecture_credit_hours
      SET lab_credit_hours TO input course lab_credit_hours
      WHEN successful
      MOVE course TO output course
      EXIT STATE IS successful_operation
      WHEN not unique
      EXIT STATE IS course_nu
  -- WHEN not found
      EXIT STATE IS course_nf
```

Process: REMOVE_COURSE_FROM_CATALOG

System Gen: This process removes a course from the available course listing. (Outside scope: Registrar)

```
REMOVE_COURSE_FROM_CATALOG
IMPORTS: ...
EXPORTS: ...
LOCALS:
ENTITY ACTIONS: ...

-- READ course
    WHERE DESIRED course number IS EQUAL TO input course
        number
    AND DESIRED course academic_department_code IS EQUAL TO
        input course academic_department_code
-- WHEN successful
    MOVE course TO output course
    DELETE course
    EXIT STATE IS successful_operation
-- WHEN not found
    EXIT STATE IS course_nf
```

Process: MODIFY_PASSWORD

System Gen: This process creates a password for a particular curricular office for use by the system to restrict the view available to a given curricular office; the operation of this process will, of course, be transparent to the users. (Outside scope: MIS)

MODIFY_PASSWORD

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

READ curricular_office

WHERE DESIRED curricular_office code IS EQUAL TO input
curricular_office code

WHEN successful

UPDATE curricular_office

SET password TO input curricular_office password

WHEN successful

MOVE curricular_office TO output curricular_office

EXIT STATE IS successful_operation

WHEN not unique

EXIT STATE IS curricular_office_nu

WHEN not found

EXIT STATE IS curricular_office_nf

Process: ASSIGN_NAVY_BOOK_CEILING

System Gen: This process creates the Naval ceiling for the Naval Book Eligibility. (Outside scope: MIS)

```

[ ASSIGN_NAVY_BOOK_CEILING
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  [ CREATE total_annual_naval_book_ceiling
    SET total_amount TO input total_annual_naval_book_ceiling
      total_amount
    SET date_implemented TO input total_annual_naval_book_ceiling
      date_implemented
  [ WHEN successful
    MOVE total_annual_naval_book_ceiling TO output
      total_annual_naval_book_ceiling
    EXIT STATE IS successful_operation
  [ WHEN already exists
    EXIT STATE IS navy_book_eligibility_ae
  ]
]
```

Process: MODIFY_NAVY_BOOK_CEILING

System Gen: This process modifies the Navy Book
Eligibility. (Outside scope: MIS)

MODIFY NAVY_BOOK_CEILING

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS: ...

-- READ total_annual_naval_book_ceiling

WHERE DESIRED total_annual_naval_book_ceiling

date_implemented IS EQUAL TO input

total_annual_naval_book_ceiling date_implemented

-- WHEN successful

UPDATE total_annual_naval_book_ceiling

SET total_amount TO input total_annual_naval_book_ceiling
total_amount

-- WHEN successful

MOVE total_annual_naval_book_ceiling TO output
total_annual_naval_book_ceiling

EXIT STATE IS successful_operation

-- WHEN not unique

EXIT STATE IS navy_book_eligibility_nu

-- WHEN not found

EXIT STATE IS navy_book_eligibility_nf

APPENDIX I

The Action Diagrams on the following pages define the logic of a derivation algorithm. These BAA Action Blocks were developed for the derived or designed attributes of the data model.

A derived attribute is one whose values can be calculated from values of other attributes and relationships. Its value can change over time as the other attributes and relationships change. A designed attribute is also calculated or deduced, but its value, once determined, does not change.

[Ref. 28:p. 8-13]

Only one attribute is set as an output of these action blocks. Since designed attributes do not change, they are SET in process action statements in Appendix H through an accessing action block mechanism called, USING. Derived attributes do not need to be SET in the process action statements since they are SET automatically when attributes, used in the calculation of the derived attribute, are changed.

The outline below indicates the attributes set by the Action Diagrams on the following pages:

AMOUNT REMAINING

Attribute: AMOUNT REMAINING of Entity Type:
STUDENT_BOOK_REIMBURSEMENT

CALCULATE GRADUATE_QPR

Attribute: GRADUATE_QPR of Entity Type: STUDENT

CALCULATE_QTR GRADUATE_QPR

Attribute: GRADUATE_QPR of Entity Type: QUARTER_QPR

CALCULATE_QTR TOTAL_QPR

Attribute: TOTAL_QPR of Entity Type: QUARTER_QPR

CALCULATE TOTAL_QPR

Attribute: TOTAL_QPR of Entity Type: STUDENT

DETERMINE NUMBER_ACADEMIC_QTRS

Attribute: NUMBER_ACADEMIC_QTRS_AUTHORIZED of Entity Type:
STUDENT_BOOK_REIMBURSEMENT

THESIS_NUMBER (only designed algorithm)

Attribute: NUMBER of Entity Type: THESIS

BAA Action Block: AMOUNT_REMAINING

```
AMOUNT_REMAINING
IMPORTS: ...
EXPORTS: ...
LOCALS:
    Work View   ief_supplied
                total_currency
ENTITY ACTIONS: ...

SET ief_supplied total_currency TO 0
-- READ EACH existing_book_claim
    WHERE DESIRED existing_book_claim reduces
        import student_book_reimbursement
SET ief_supplied total_currency TO (ief_supplied total_currency
    + existing_book_claim amount_of_claim)

SET export student_book_reimbursement amount_remaining TO
    (student_book_reimbursement total_amount_eligible -
    ief_supplied total_currency)
```

BAA Action Block: CALCULATE_GRADUATE_QPR

CALCULATE_GRADUATE_QPR

IMPORTS: ...

EXPORTS: ...

LOCALS:

Work View temp qpr
total

Work View credits gained
cumulative_hours

Work View grade point
value

cumulative_value

ENTITY ACTIONS: ...

READ EACH student_course_of_study

WHERE DESIRED student_course_of_study belongs_to
input student

READ course

WHERE DESIRED course assigned to
CURRENT student_course_of_study

WHEN successful

IF course number IS GREATER OR EQUAL TO 3000

CASE OF student_course_of_study grade

CASE "A"

SET grade point value TO 4

CASE "A-"

SET grade point value TO 3.7

CASE "B+"

SET grade point value TO 3.3

CASE "B"

SET grade point value TO 3

CASE "B-"

SET grade point value TO 2.7

CASE "C+"

SET grade point value TO 2.3

CASE "C"

SET grade point value TO 2

CASE "C-"

SET grade point value TO 1.7

CASE "D+"

SET grade point value TO 1.3

CASE "D"

SET grade point value TO 1

OTHERWISE

SET grade point value TO 0

SET grade point value TO (grade point value * course
lecture_credit_hours)

```
SET credits_gained_cumulative_hours TO (credits_gained  
cumulative_hours + course  
lecture_credit_hours)  
SET grade_point_cumulative_value TO (grade_point  
cumulative_value + grade_point_value)  
SET output_student_graduate_qpr TO (grade_point  
cumulative_value / credits_gained  
cumulative_hours)
```

```
WHEN not found  
EXIT STATE IS course_nf
```

BAA Action Block: CALCULATE_QTR_GRADUATE_QPR

CALCULATE_QTR_GRADUATE_QPR

IMPORTS: ...

EXPORTS: ...

LOCALS:

Work View temp qpr
total

Work View credits gained
cumulative_hours

Work View grade point
value
cumulative_value

ENTITY ACTIONS: ...

READ EACH student_course_of_study

WHERE DESIRED student_course_of_study used_to_calculate
input quarter_qpr

READ course

WHERE DESIRED course assigned_to
CURRENT student_course_of_study

WHEN successful

IF course number IS GREATER OR EQUAL TO 3000

CASE OF student_course_of_study grade

CASE "A"

SET grade point value TO 4

CASE "A-"

SET grade point value TO 3.7

CASE "B+"

SET grade point value TO 3.3

CASE "B"

SET grade point value TO 3

CASE "B-"

SET grade point value TO 2.7

CASE "C+"

SET grade point value TO 2.3

CASE "C"

SET grade point value TO 2

CASE "C-"

SET grade point value TO 1.7

CASE "D+"

SET grade point value TO 1.3

CASE "D"

SET grade point value TO 1

OTHERWISE

SET grade point value TO 0

SET grade point value TO (grade point value * course
lecture_credit_hours)

```
SET credits_gained_cumulative_hours TO (credits_gained  
cumulative_hours + course  
lecture_credit_hours)  
SET grade_point_cumulative_value TO (grade_point  
cumulative_value + grade_point_value)  
SET output_quarter_qpr_graduate TO (grade_point  
cumulative_value / credits_gained  
cumulative_hours)
```

```
WHEN not_found  
EXIT STATE IS course_nf
```

BAA Action Block: CALCULATE_QTR_TOTAL_QPR

CALCULATE_QTR_TOTAL_QPR

IMPORTS: ...

EXPORTS: ...

LOCALS:

Work View temp qpr
total

Work View credits gained
cumulative_hours

Work View grade point
value
cumulative_value

ENTITY ACTIONS: ...

READ EACH student_course_of_study

WHERE DESIRED student_course_of_study used_to_calculate
input quarter_qpr

READ course

WHERE DESIRED course assigned to
CURRENT student_course_of_study

WHEN successful

CASE OF student_course_of_study grade

CASE "A"

SET grade point value TO 4

CASE "A-"

SET grade point value TO 3.7

CASE "B+"

SET grade point value TO 3.3

CASE "B"

SET grade point value TO 3

CASE "B-"

SET grade point value TO 2.7

CASE "C+"

SET grade point value TO 2.3

CASE "C"

SET grade point value TO 2

CASE "C-"

SET grade point value TO 1.7

CASE "D+"

SET grade point value TO 1.3

CASE "D"

SET grade point value TO 1

OTHERWISE

SET grade point value TO 0

SET grade point value TO (grade point value * course
lecture_credit_hours)

```
SET credits_gained_cumulative_hours TO (credits_gained  
cumulative_hours + course_lecture_credit_hours)  
SET grade_point_cumulative_value TO (grade_point  
cumulative_value + grade_point_value)  
SET output_quarter_qpr_total TO (grade_point  
cumulative_value / credits_gained  
cumulative_hours)  
- WHEN not_found  
EXIT STATE IS course_nf
```


BAA Action Block: CALCULATE_TOTAL_QPR

CALCULATE_TOTAL_QPR

IMPORTS: ...

EXPORTS: ...

LOCALS:

Work View temp qpr

total

Work View credits gained

cumulative_hours

Work View grade point

value

cumulative_value

ENTITY ACTIONS: ...

READ EACH student_course_of_study

WHERE DESIRED student_course_of_study belongs_to

import student

READ course

WHERE DESIRED course assigned_to

CURRENT student_course_of_study

WHEN successful

CASE OF student_course_of_study grade

CASE "A"

SET grade point value TO 4

CASE "A-"

SET grade point value TO 3.7

CASE "B+"

SET grade point value TO 3.3

CASE "B"

SET grade point value TO 3

CASE "B-"

SET grade point value TO 2.7

CASE "C+"

SET grade point value TO 2.3

CASE "C"

SET grade point value TO 2

CASE "C-"

SET grade point value TO 1.7

CASE "D+"

SET grade point value TO 1.3

CASE "D"

SET grade point value TO 1

OTHERWISE

SET grade point value TO 0

SET grade point value TO (grade point value * course
lecture_credit_hours)

```

SET credits_gained_cumulative_hours TO (credits_gained
                                         cumulative_hours + course_lecture_credit_hours)
SET grade_point_cumulative_value TO (grade_point
                                      cumulative_value + grade_point_value)
SET output_student_total_qpr TO (grade_point
                                 cumulative_value / credits_gained
                                 cumulative_hours)
WHEN not found
EXIT STATE IS course_nf

```

```

- DETERMINE_NUMBER_ACADEMIC_QTRS
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
    Work View    date
    graduation_year
    incoming_year
    graduation_month
    incoming_month
  ENTITY ACTIONS: ...

- READ student_book_reimbursement
- WHEN successful
  - READ navy
    WHERE DESIRED navy obtains
      CURRENT student_book_reimbursement
  - WHEN successful
    SET date incoming_month TO month(navy convening_date)
    SET date graduation_month TO month(navy
      anticipated_graduation_date)
    [ IF date incoming_month IS LESS THAN 10
      SET date incoming_year TO year(navy convening_date)
    ]
    [ ELSE
      SET date incoming_year TO year(navy convening_date) + 1
    ]
    [ IF date graduation_month IS LESS THAN 10
      SET date graduation_year TO year(navy
        anticipated_graduation_date)
    ]
    [ ELSE
      SET date graduation_year TO year(navy
        anticipated_graduation_date) + 1
    ]
    [ IF student_book_reimbursement year IS EQUAL TO date
      incoming_year
      [ IF date incoming_month IS GREATER OR EQUAL TO 10
        SET export student_book_reimbursement
          number_academic_qtrs_authorized TO 1
      ]
      [ ELSE IF date incoming_month IS GREATER OR EQUAL TO 7
        SET export student_book_reimbursement
          number_academic_qtrs_authorized TO 4
      ]
      [ ELSE IF date incoming_month IS GREATER OR EQUAL TO 4
        SET export student_book_reimbursement
          number_academic_qtrs_authorized TO 2
      ]
    ]
  ]

```

```

      ELSE
      SET export student_book_reimbursement
        number_academic_qtrs_authorized TO 3
    ELSE IF student_book_reimbursement year IS EQUAL TO date
      graduation_year
      IF date graduation_month IS GREATER OR EQUAL TO 10
      SET export student_book_reimbursement
        number_academic_qtrs_authorized TO 4
      ELSE IF date graduation_month IS GREATER OR EQUAL TO 7
      SET export student_book_reimbursement
        number_academic_qtrs_authorized TO 1
      ELSE IF date graduation_month IS GREATER OR EQUAL TO 4
      SET export student_book_reimbursement
        number_academic_qtrs_authorized TO 3
      ELSE
      SET export student_book_reimbursement
        number_academic_qtrs_authorized TO 2
    ELSE
    SET export student_book_reimbursement
      number_academic_qtrs_authorized TO 4
  WHEN not found
WHEN not found

READ navy
  WHERE DESIRED navy obtains
    CURRENT student_book_reimbursement
  WHEN successful
  SET date incoming_month TO month(navy convening_date)
  SET date graduation_month TO month(navy
    anticipated_graduation_date)
    IF date incoming_month IS LESS THAN 10
    SET date incoming_year TO year(navy convening_date)
    ELSE
    SET date incoming_year TO year(navy convening_date) + 1
    IF date graduation_month IS LESS THAN 10
    SET date graduation_year TO year(navy
      anticipated_graduation_date)
    ELSE
    SET date graduation_year TO year(navy
      anticipated_graduation_date) + 1

```

```

IF student_book_reimbursement_year IS EQUAL TO date
    incoming_year
    IF date_incoming_month IS GREATER OR EQUAL TO 10
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 4
    ELSE IF date_incoming_month IS GREATER OR EQUAL TO 7
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 1
    ELSE IF date_incoming_month IS GREATER OR EQUAL TO 4
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 2
    ELSE
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 3
ELSE IF student_book_reimbursement_year IS EQUAL TO date
    graduation_year
    IF date_graduation_month IS GREATER OR EQUAL TO 10
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 4
    ELSE IF date_graduation_month IS GREATER OR EQUAL TO 7
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 1
    ELSE IF date_graduation_month IS GREATER OR EQUAL TO 4
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 2
    ELSE
        SET export student_book_reimbursement
            number_academic_qtrs_authorized TO 3
ELSE
    SET export student_book_reimbursement
        number_academic_qtrs_authorized TO 4
WHEN not found
EXIT STATE IS student_book_money_nf

```

BAA Action Block: THESIS_NUMBER

```

[ THESIS_NUMBER
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS: ...

  SET output thesis number TO 1
  [ READ EACH thesis
    SORTED BY DESCENDING thesis number
    WHERE DESIRED thesis due_date_year IS EQUAL TO input
      thesis due_date_year
    SET output thesis number TO (thesis number + 1)
  [←ESCAPE
  ]
]
```

APPENDIX J

The Process Dependency Diagrams on the following pages document the sequence in which processes must occur. This sequence is based on dependencies between functions/processes, including logic and timing constraints. It also shows the source of information required by the processes and the destination of information produced by the processes. [Ref. 29:p. 23] and [Ref. 5:p. 8-27]

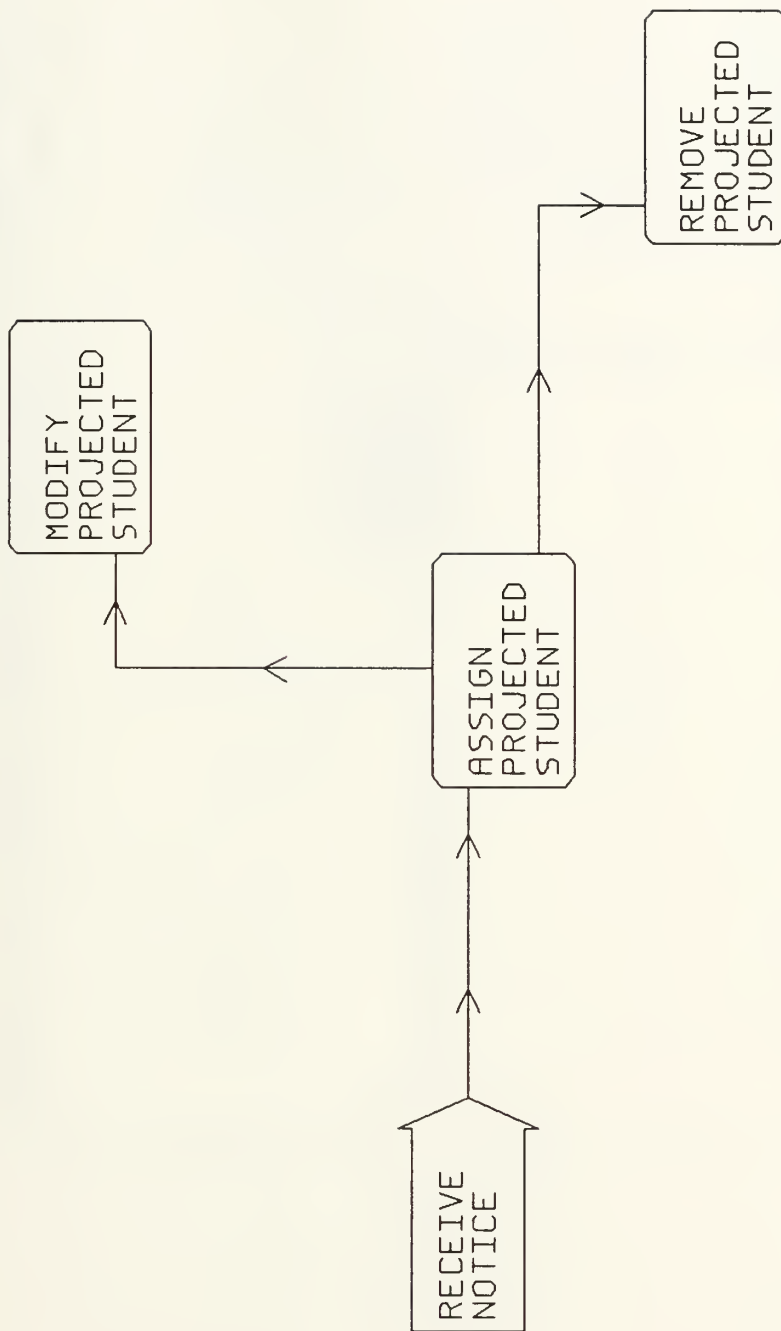
Large labeled arrows depict events which are a point in time relevant to a process; the passing of a specific point in time that triggers the execution of one or more processes.

Layered boxes represent external objects which provide data to a process and/or receive results from a process.

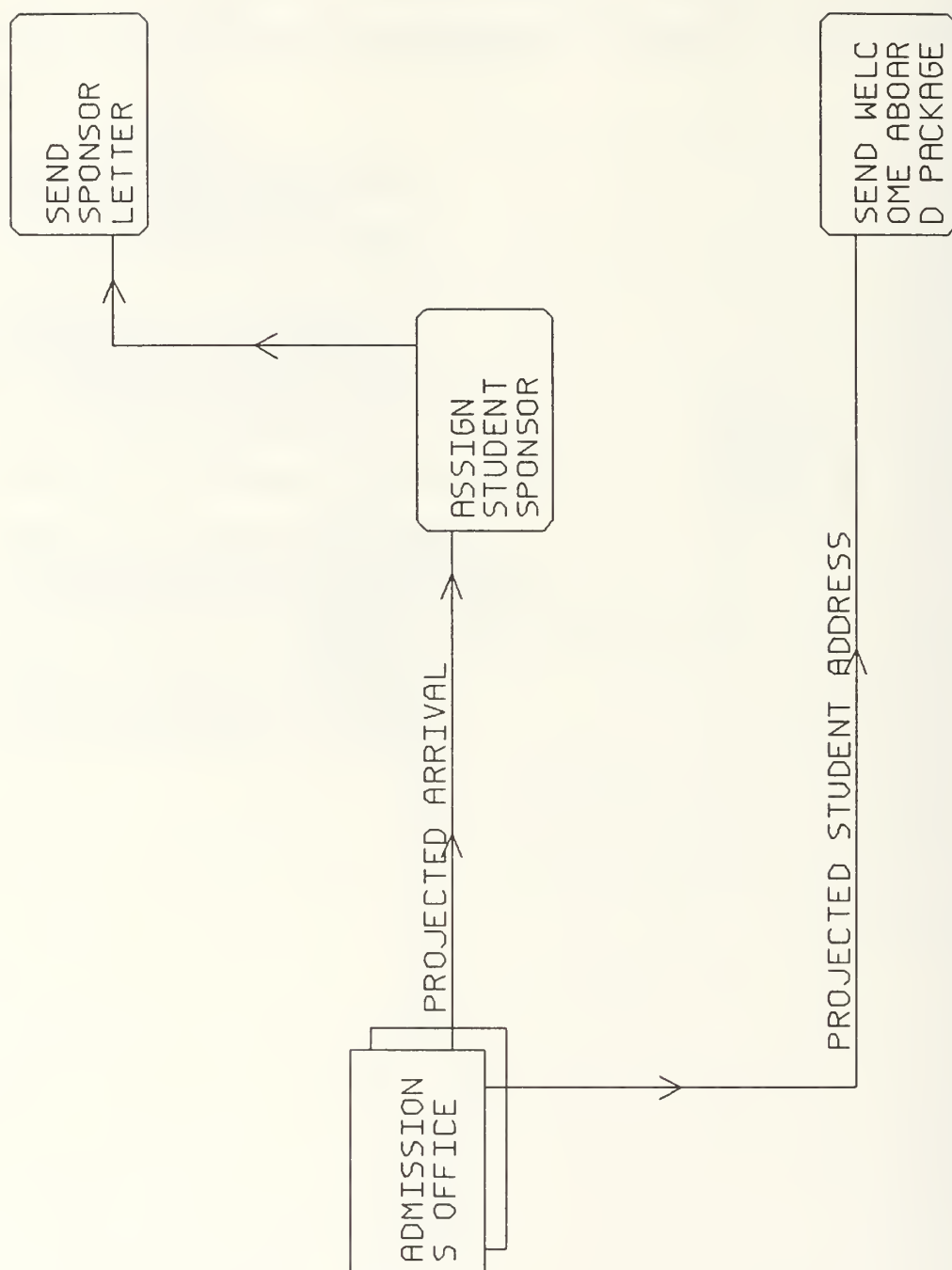
Rounded boxes represent the process which modifies data in some manner.

The outline below indicates those functions or high-level processes which possess Dependency Diagrams on the following pages:

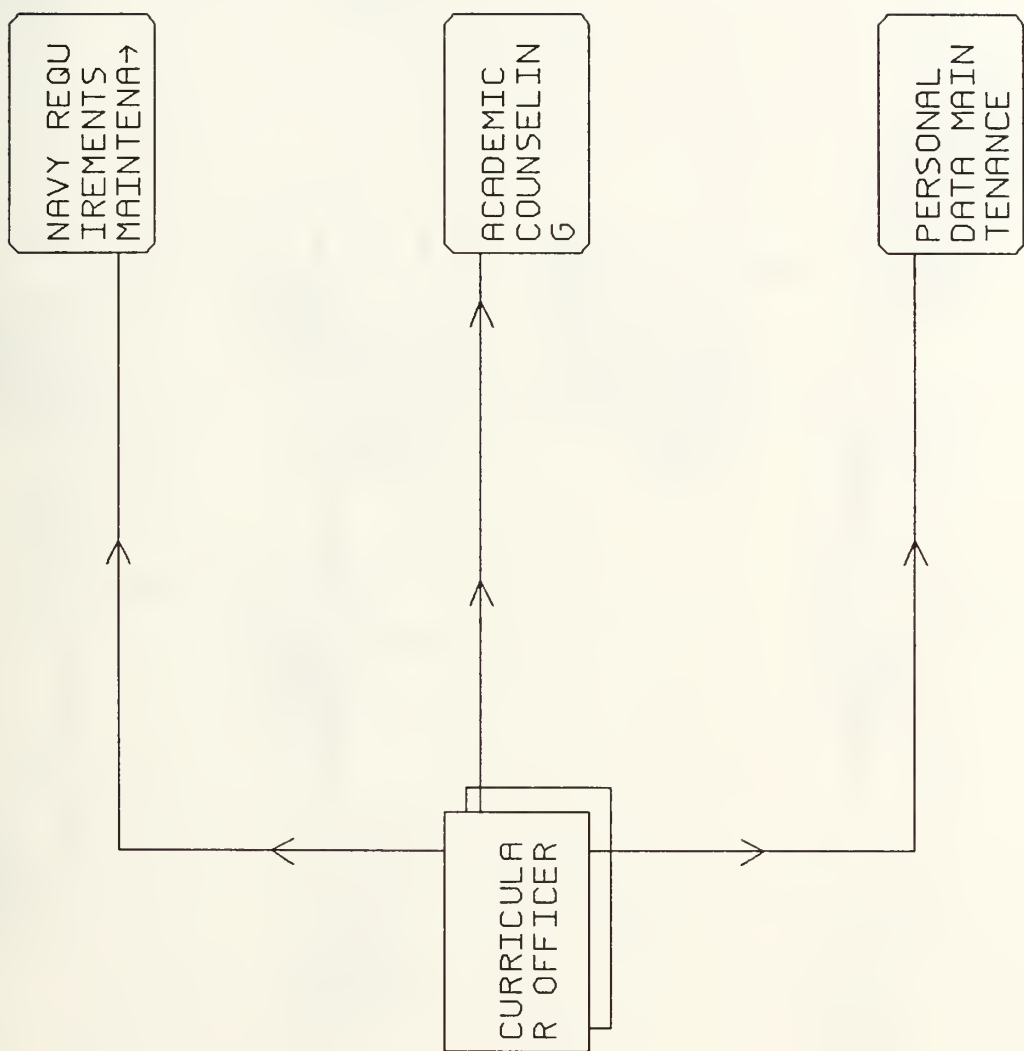
- 1 INITIALIZE_STUDENT_RECORD
- 2 COUNSELING_FUTURE_STUDENT
- 3 SUPERVISE_ENROLLED_STUDENT
 - 3.1 ACADEMIC_COUNSELING
 - 3.2 PERSONAL_DATA_MAINTENANCE
 - 3.2.7 RECORD_STUDENT_DATA
 - 3.3 NAVY_REQUIREMENTS_MAINTENANCE
- 4 COMPLETED_ACADEMIC_REQUIREMENTS
- 5 CURRICULUM_DEVELOPMENT_MGMT
 - 5.1 CURRICULUM_OFFICE_MAINTENANCE
 - 5.2 COURSE_OF_STUDY_MAINTENANCE
- 6 COURSE_MAINTENANCE
- 7 SYSTEM_MANAGEMENT



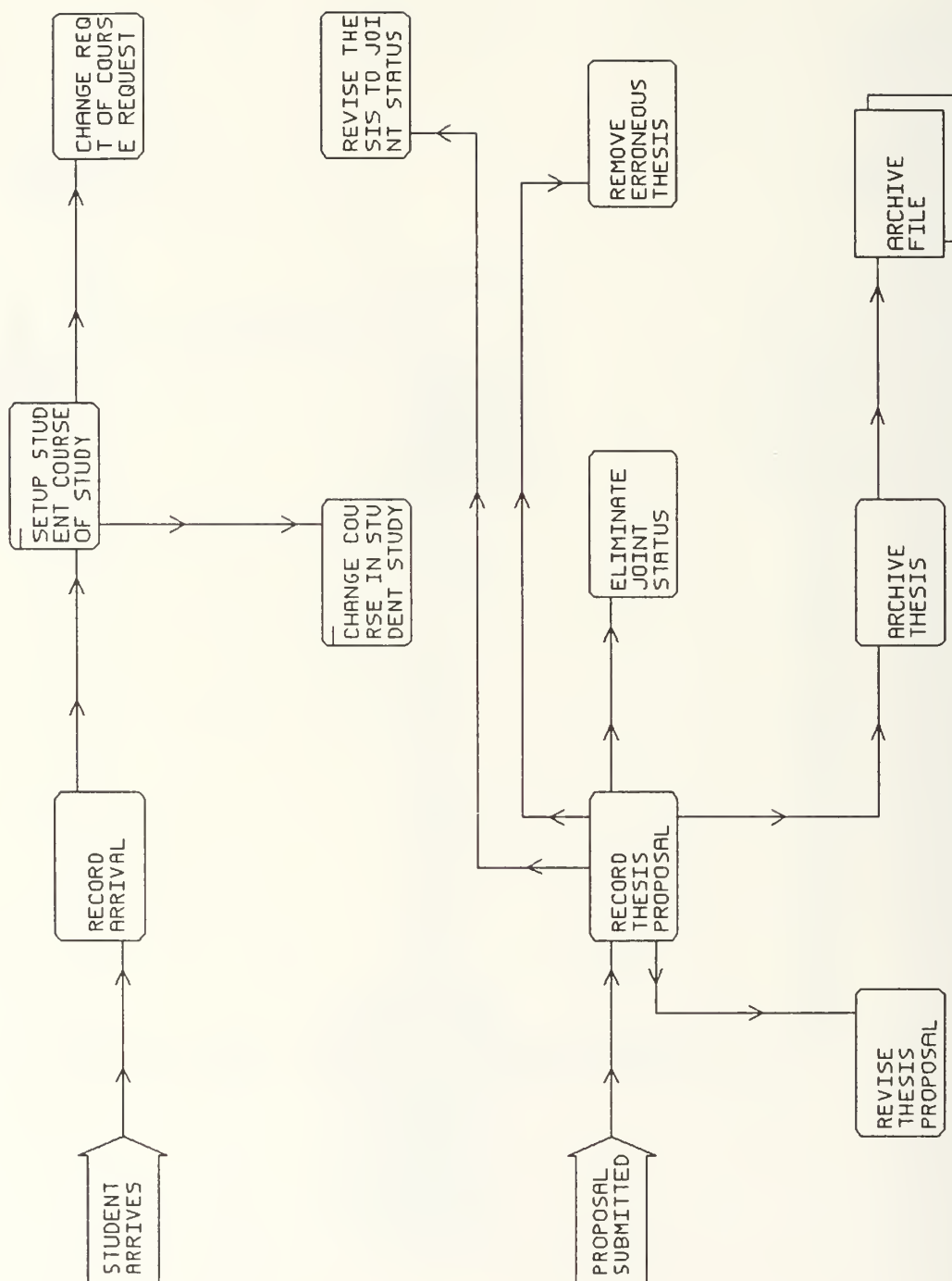
INITIALIZE STUDENT RECORD



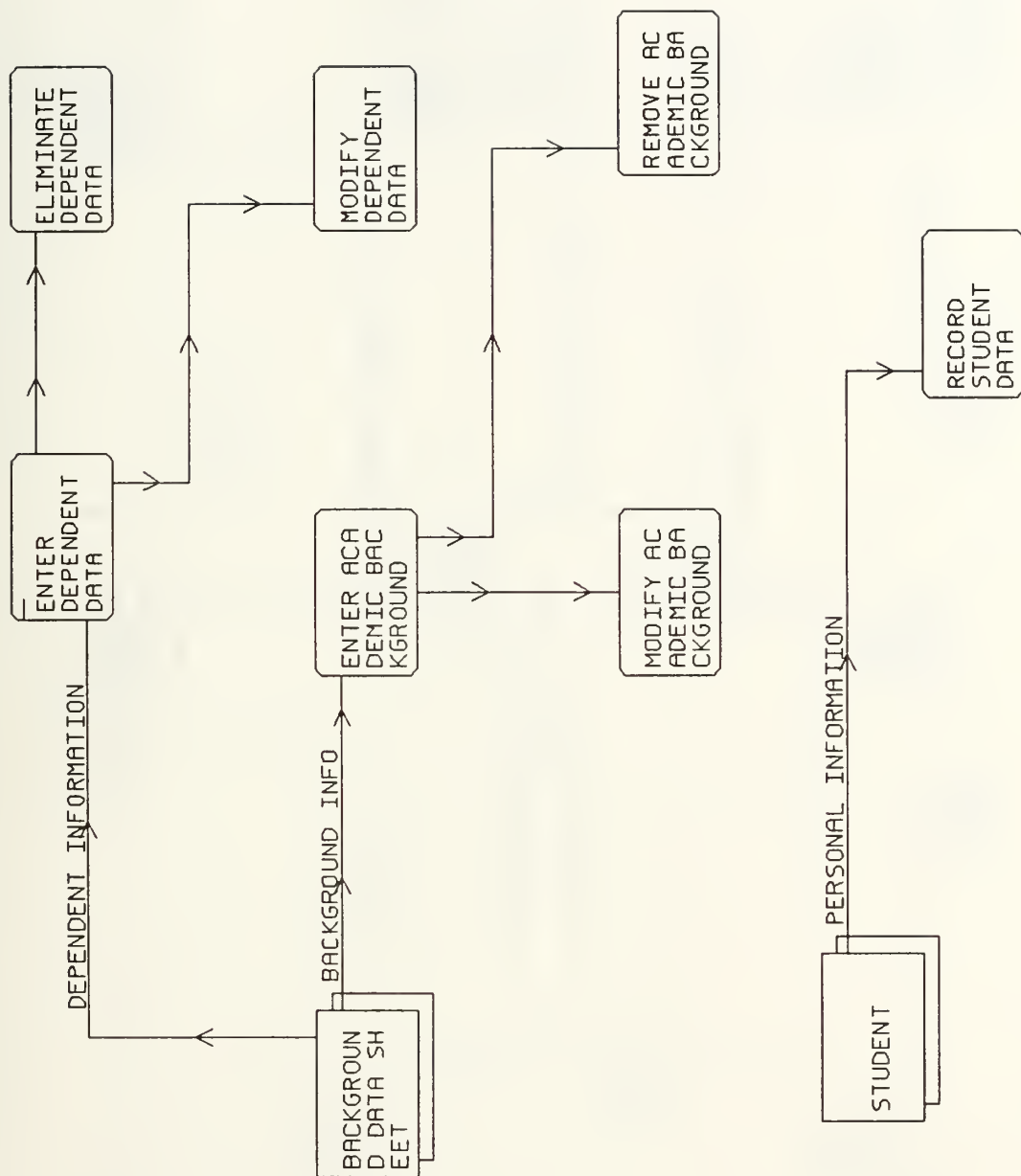
COUNSELING FUTURE STUDENT



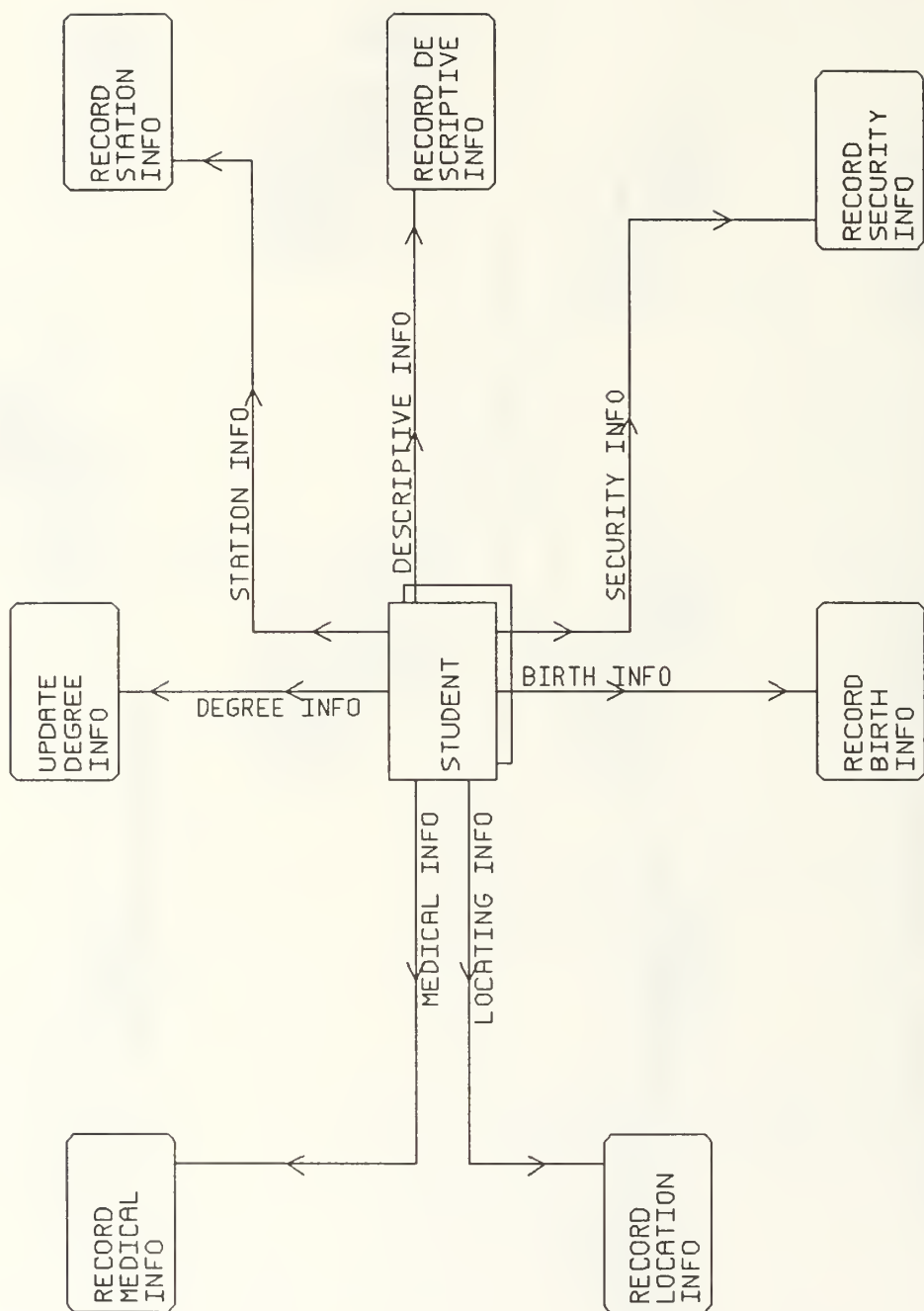
SUPERVISE ENROLLED STUDENT



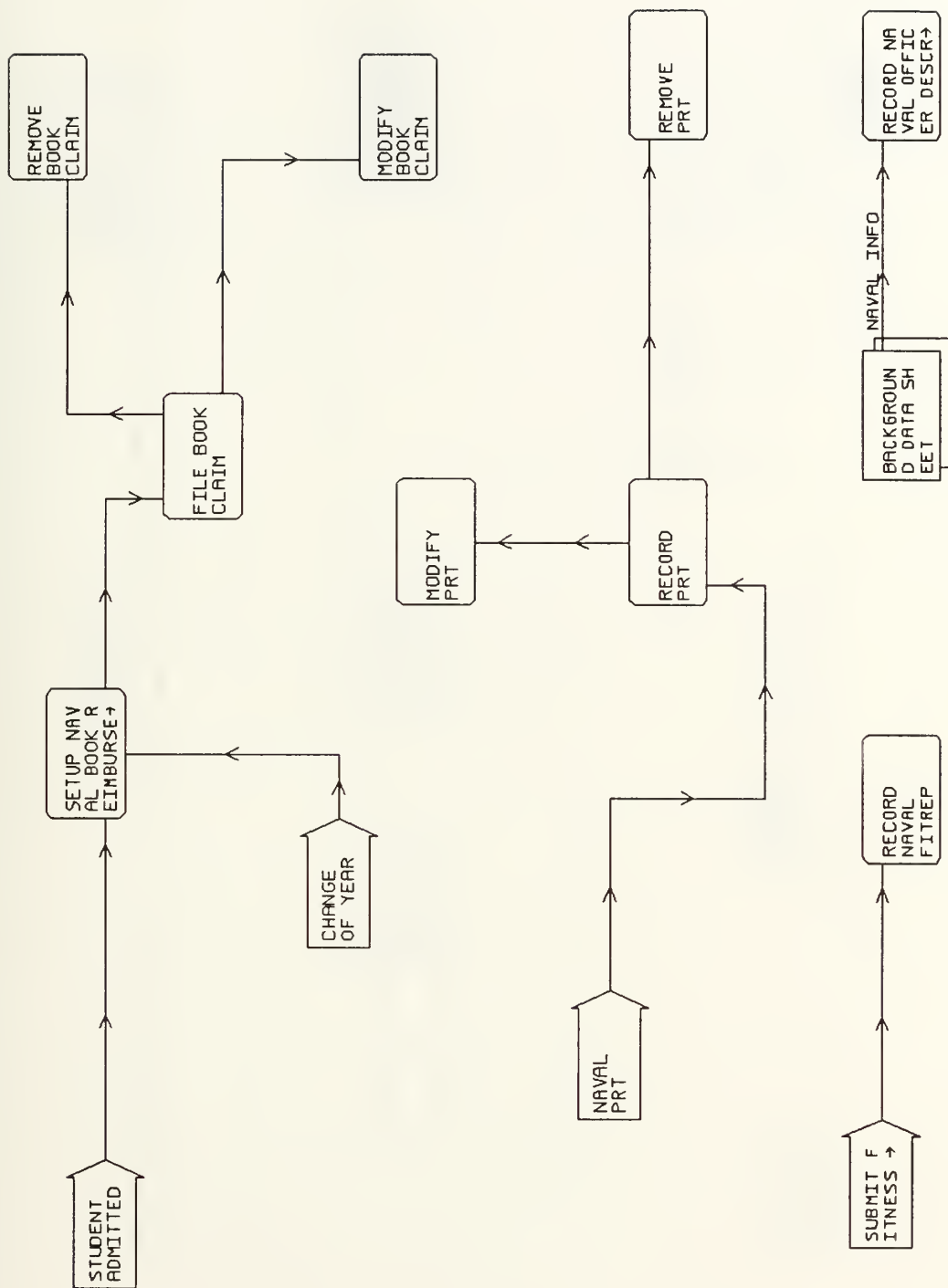
ACADEMIC COUNSELING



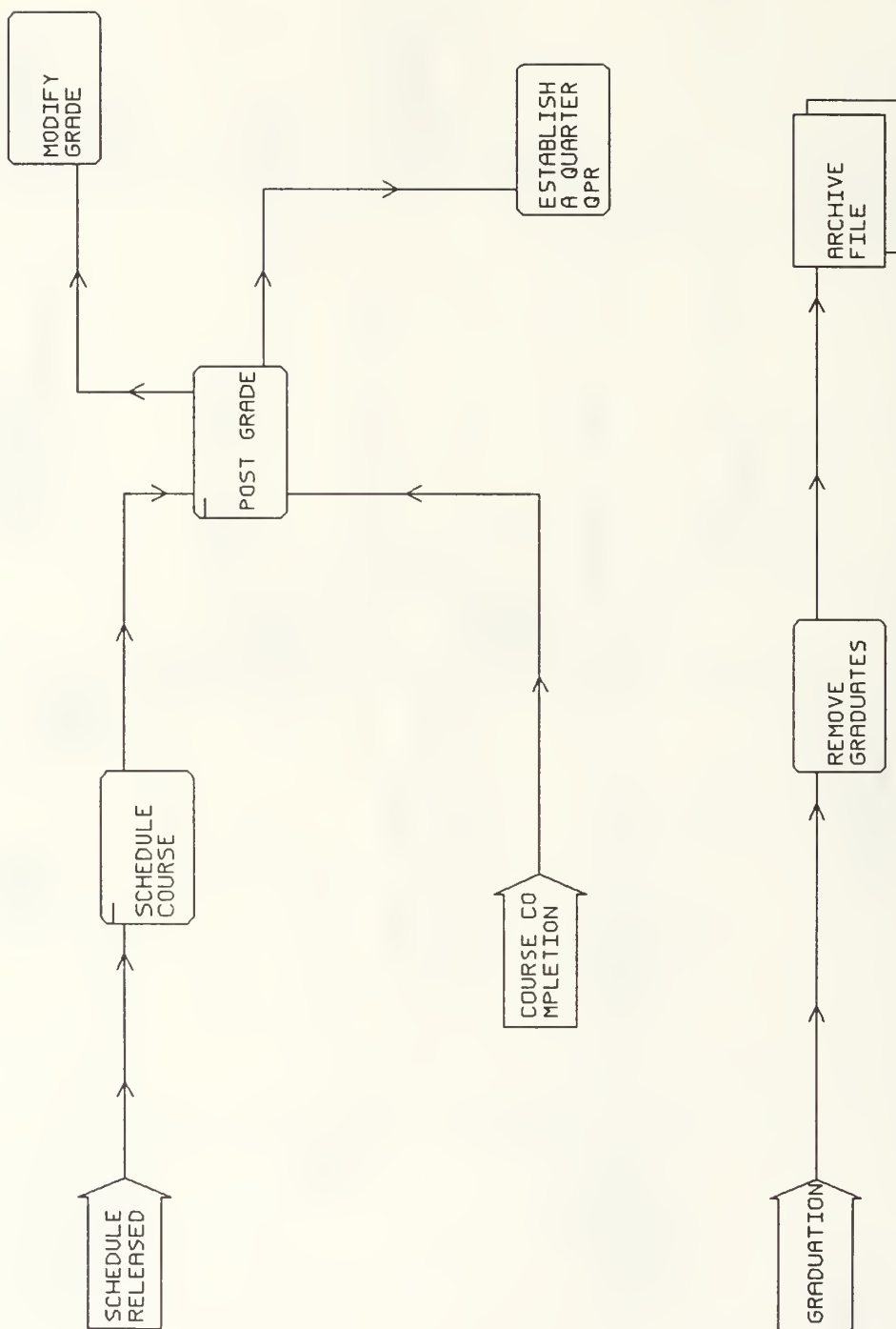
PERSONAL DATA MAINTENANCE



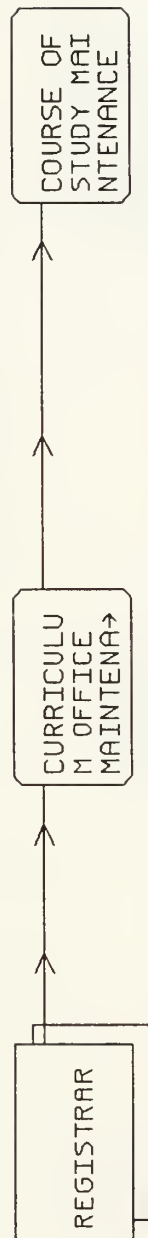
RECORD STUDENT DATA



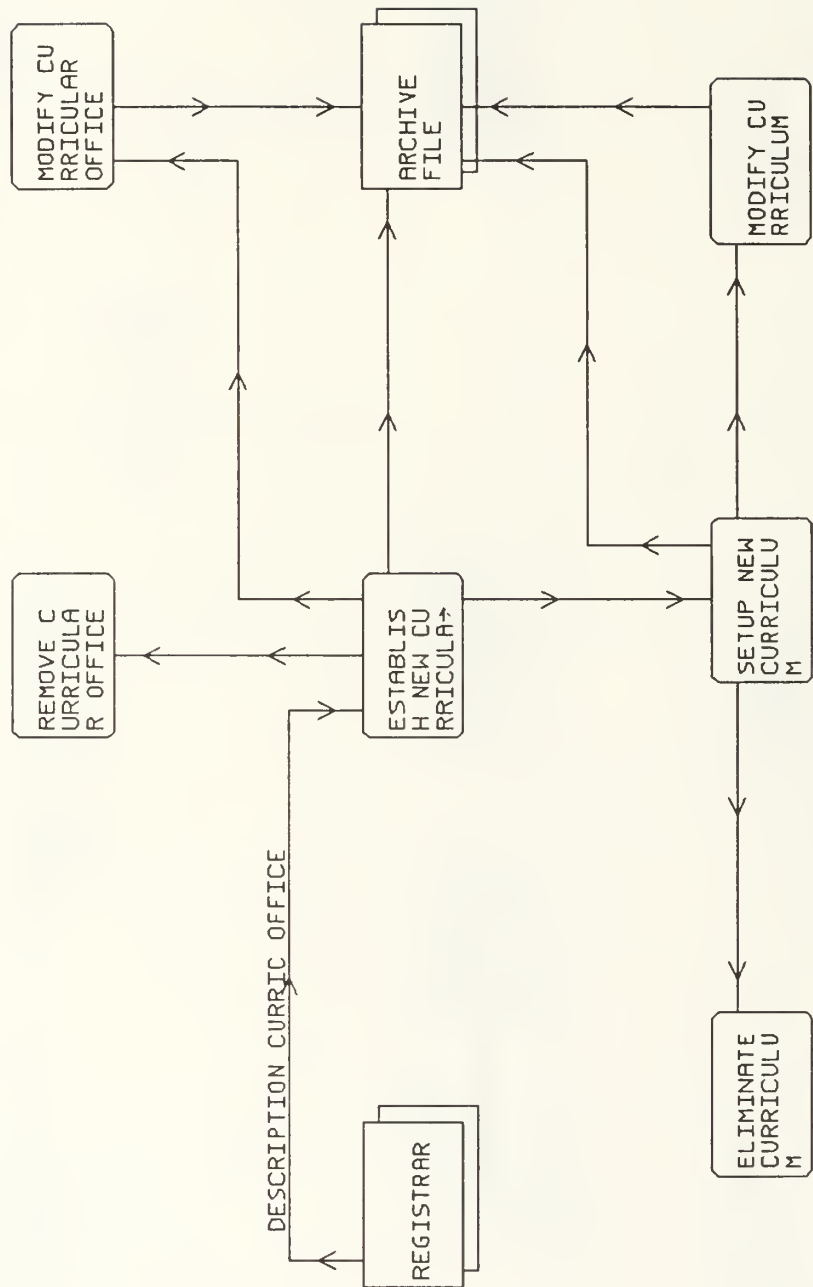
NAVY REQUIREMENTS MAINTENANCE



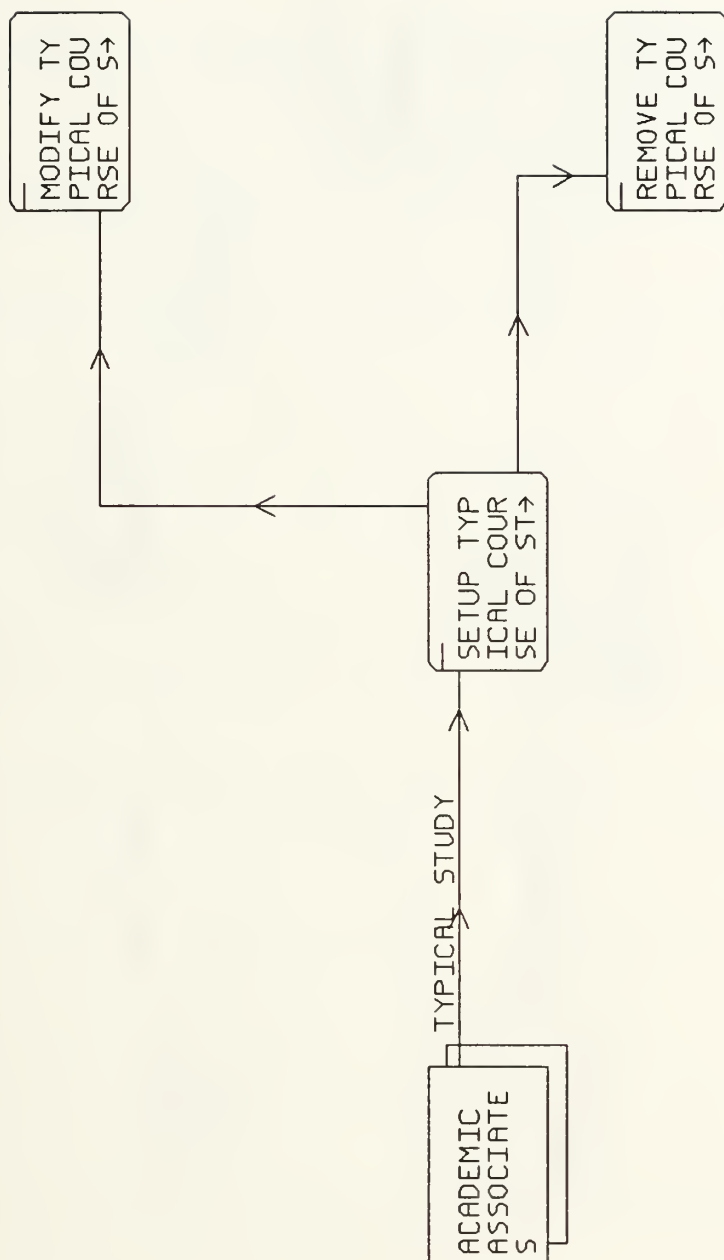
COMPLETED ACADEMIC REQUIREMENTS



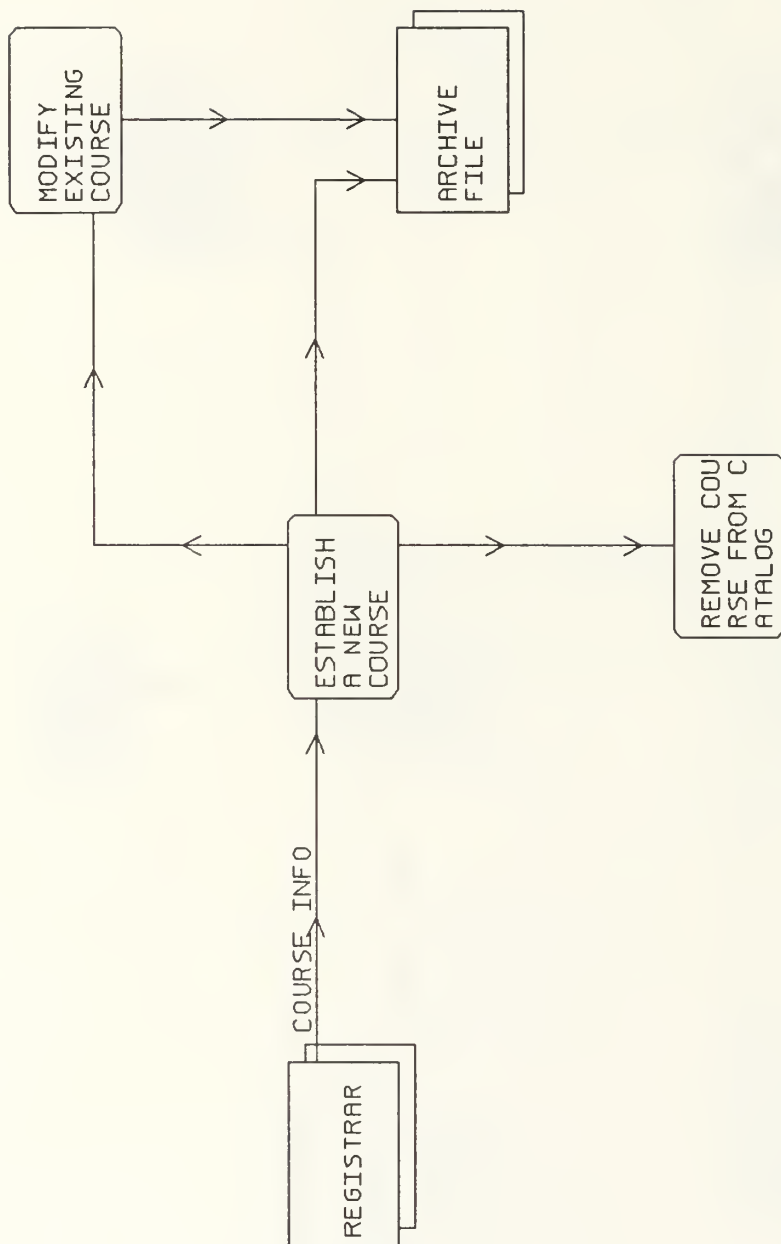
CURRICULUM DEVELOPMENT MGMT



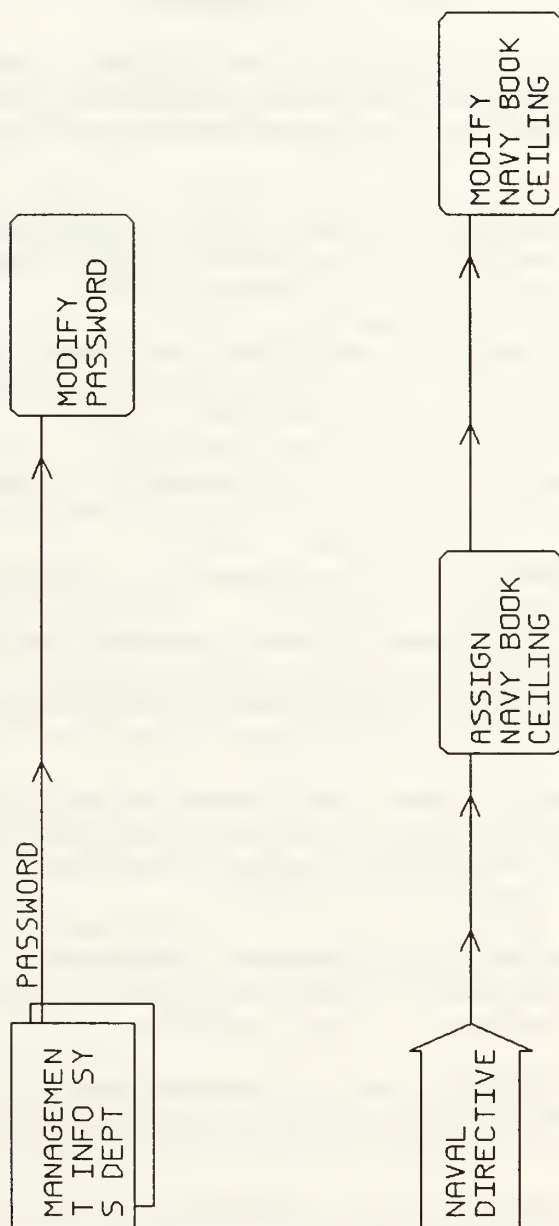
CURRICULUM OFFICE MAINTENANCE



COURSE OF STUDY MAINTENANCE



COURSE MAINTENANCE



SYSTEM MANAGEMENT

LIST OF REFERENCES

1. Associate Dean of Faculty and Graduate Studies (07/Er), Naval Postgraduate School, Monterey, CA Memorandum, Subject: Unification of Academic Databases, 27 February 1990.
2. Interview between B. Frew, Dean, Dean of Computers and Information Services, Naval Postgraduate School, Monterey, CA, and the authors, 24 October 1990.
3. Interview between M. Spencer, Director, Management of Information Systems, Naval Postgraduate School, Monterey, CA, and the authors, 21 February 1991.
4. Martin, James and Finkelstein, Clive, **Information Engineering**, Vols 1 and 2, Savant Institute, 1982, p. 2.
5. Texas Instruments Incorporated, **A Guide to Information Engineering Using the IEF: Computer-Aided Planning, Analysis, and Design**, 2nd ed., January 1990.
6. Texas Instrument Incorporated, **Information Engineering Facility Methodology Overview**, 1990.
7. Rouska, A. M., LT, USN, and VanNortwick, E. D., LCDR, USN, "Curricular Officer Student Information System Requirements Analysis," study prepared at the Naval Postgraduate School, May 1990.
8. Interview between M. Spencer, Director, Management of Information Systems, Naval Postgraduate School, Monterey, CA, and the authors, 3 April 1991.
9. Gear, Mary, "FOCUS - the Database System of the Curricular Offices at NPS", paper presented to Naval Postgraduate School, Monterey, CA, 10 May 1989.
10. Interview between T. Hoskins, CDR, USN, Computer Technology Curricular Officer, Naval Postgraduate School, Monterey, CA, and the authors, 3 October 1990.

11. Telephone conversation between M. Spencer, Director, Management of Information Systems, Naval Postgraduate School, Monterey, CA, and one of the authors, 14 December 1990.
12. Interview between T. Hoskins, CDR, USN, Computer Technology Curricular Officer, Naval Postgraduate School, Monterey, CA, and the authors, 29 May 1991.
13. Uluakar, Tamer, "From Structured Methods to Information Engineering: A Comparison", paper presented to Texas Instruments, Inc., Iselin, New Jersey, March 1991.
14. Woodburn, D., "I.S. Stereotypes Change with CASE," **Canadian Information Processing**, pp. 10-13, December 1990/January 1991.
15. Senn, J.A., **Analysis and Design of Information Systems**, McGraw-Hill Publishing Company, 1989.
16. Sharon, D., "Look Beyond the 'I CASE' Label," **Computerworld**, Vol. XXV, No. 16, pp. 61-63, 22 April 1991.
17. Gartner Group, Incorporated Strategic Planning Report SPA-960-484, **Software Engineering Strategies, Comparing CASE Solution Productivity**, by A. Case, A. Rin, 30 November 1990.
18. Sullivan-Trainor, M. L., "TI's IEF Scores High for Integration, Benefits Delivery," **Computerworld**, Vol. XXV, No. 16, pp. 72-73, 22 April 1991.
19. Interview between T. King, Product Specialist, and J. Penrod, Product Specialist, Texas Instruments Incorporated, San Francisco, CA, and the authors, 12 April 1991.
20. Texas Instruments Incorporated, **IEF Rapid Development/Tutorial Module, Beta Version - V0.5**, 11 February 1991.
21. Telephone conversation between L.E. Rowland, Department Head of the Information Systems Department (Code 422), Naval Aviation Management Office, Naval Air Station Patuxent River, MD, 17 July 1991.

22. Telephone conversation between J. Joseph, Computer Systems Analyst, Naval Aviation Maintenance Office, Naval Air Station Patuxent River, MD, and one of the authors, 8 July 1991.
23. Telephone conversation between J. Albenesius, Technical Project Coordinator, Federal Reserve Bank, St. Louis, MO, and one of the authors, 24 May 1991.
24. Telephone conversation between J. Penrod, Product Specialist, Texas Instruments Incorporated, Santa Clara, CA, and one of the authors, 17 July 1991.
25. Interview between M. Spencer, Director, Management of Information Systems, Naval Postgraduate School, Monterey, CA, and the authors, 26 April 1991.
26. Texas Instruments Incorporated, IEF, Information Engineering Facility, Planning Toolset Guide, 1990.
27. Texas Instruments Incorporated, IEF, Information Engineering Facility, Analysis Toolset Guide, 1990.
28. Texas Instruments Incorporated, IEF, Information Engineering Facility, IEF Basics, 1st ed., December 1989.
29. Texas Instruments Incorporated, Information Engineering Facility Technology Overview, 1989.

INITIAL DISTRIBUTION LIST

- | | | |
|----|---|---|
| 1. | Library, Code 52
Naval Postgraduate School
Monterey, California 93943-5002 | 2 |
| 2. | Dean Barry A. Frew, Code 05
Dean of Computers and Information Services
Naval Postgraduate School
Monterey, California 93943-5000 | 2 |
| 3. | Professor Daniel R. Dolk, Code AS/Dk
Department of Administrative Sciences
Naval Postgraduate School
Monterey, California 93943-5000 | 2 |
| 4. | Professor Magdi Kamel, Code AS/Ka
Department of Administrative Sciences
Naval Postgraduate School
Monterey, California 93943-5000 | 1 |
| 5. | Captain Michael S. Haas, USMC
Headquarters Marine Corps (MCC 010)
HQBN, HQMC, Henderson Hall
Arlington, VA 22214 | 2 |
| 6. | Captain Mary L. Hochstetler, USMC
Headquarters Marine Corps (MCC 008)
HQBN, HQMC, Henderson Hall
Arlington, VA 22214 | 2 |
| 7. | Ms. Jeffrie Penrod
Texas Instruments
Info Tech Group/ IEF Group
5353 Betsy Ross Drive
Santa Clara, CA 95054 | 1 |
| 8. | Mr. Michael Spencer, Code 53
Director, Management of Information Systems
Naval Postgraduate School
Monterey, California 93943-5000 | 1 |

9. Mr. Jim Albenesius 1
Technical Project Coordinator
Information Systems Department
P.O. Box 442
St. Louis, MO 63166
10. Mr. L. E. Rowland 1
Division Head of Information Systems Department (422)
Naval Aviation Maintenance Office (NAMO)
NAS Patuxent River, MD 20670
11. Mr. Joe Joseph 1
Computer Systems Analyst
Naval Aviation Maintenance Office (NAMO)
NAS Patuxent River, MD 20670
12. Commander T. Hoskins, Code 37 1
Computer Technology Curricular Officer
Naval Postgraduate School
Monterey, California 93943-5000

Thesis

H1034 Haas

c.1 **Information Engineering**
of the curricular officers'
segment of a unified stu-
dent academic database for
NPS.

DUDLEY KNOX LIBRARY



3 2768 00033276 1